CR-128657

NOVEMBER 1972

MDC G4287

PERFORMANCE ANALYSIS AND DESIGN SYNTHESIS (PADS) COMPUTER PROGRAM

VOLUME II
Program Description
Part 2
Final Report

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY

MCDONNELL DOUGLAS CORPORATION

(NASA-CR-128657) PERFORMANCE ANALYSIS AND DESIGN SYNTHESIS (PADS) COMPUTER PROGRAM. VOLUME 2: PROGRAM DESCRIPTION, PART 2 Final (McDonnell-Douglas Astronautics Co.) 889 p HC $46.50 CSCL 22A

N73-18832

Unclas
G3/30 50669

MCDONNELL DOUGLAS

PERFORMANCE ANALYSIS AND DESIGN SYNTHESIS (PADS) COMPUTER PROGRAM
VOLUME II
Program Description
Part 2
Final Report

NOVEMBER 1972

MDC G4287

PREPARED UNDER CONTRACT NO. NAS9-12059
BY GUIDANCE AND FLIGHT MECHANICS DEPARTMENT
RESEARCH AND DEVELOPMENT
MCDONNELL DOUGLAS ASTRONAUTICS COMPANY–WEST
HUNTINGTON BEACH, CALIFORNIA
FOR
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY-WEST

5301 Bolsa Avenue, Huntington Beach, CA 92647

Quasi-Linearization Trajectory Module (GRØPE) 6

6 GRØPE

INTRODUCTION

This section of Volume II is devoted to the QL module of PADS. Execution of this module is initiated when and if subroutine PADS1 calls subroutine GROPE. Subroutine GROPE controls the high level logical flow of the QL module.

The purpose of the module is to determine a trajectory that satisfies the necessary variational conditions for optimal performance. As is shown in Section 16 of Volume I, the QL module achieves this end by solving a non-linear multi-point boundary value problem. The numerical method employed, which is alternately known as quasi-linearization or the generalized Newton-Raphson operator, is described in Section 17 of Volume I. It is an iterative technique that converges quadratically when it does converge.

The module consists of three basic steps: (1) initialization, (2) iteration, and (3) culmination. The second step has two distinct components: (A) integration of the particular and homogeneous solutions, and (B) satisfaction of the boundary conditions. Both of these components are executed for each iteration in step 2.

The FORTRAN names of the primary entry points to these steps are:

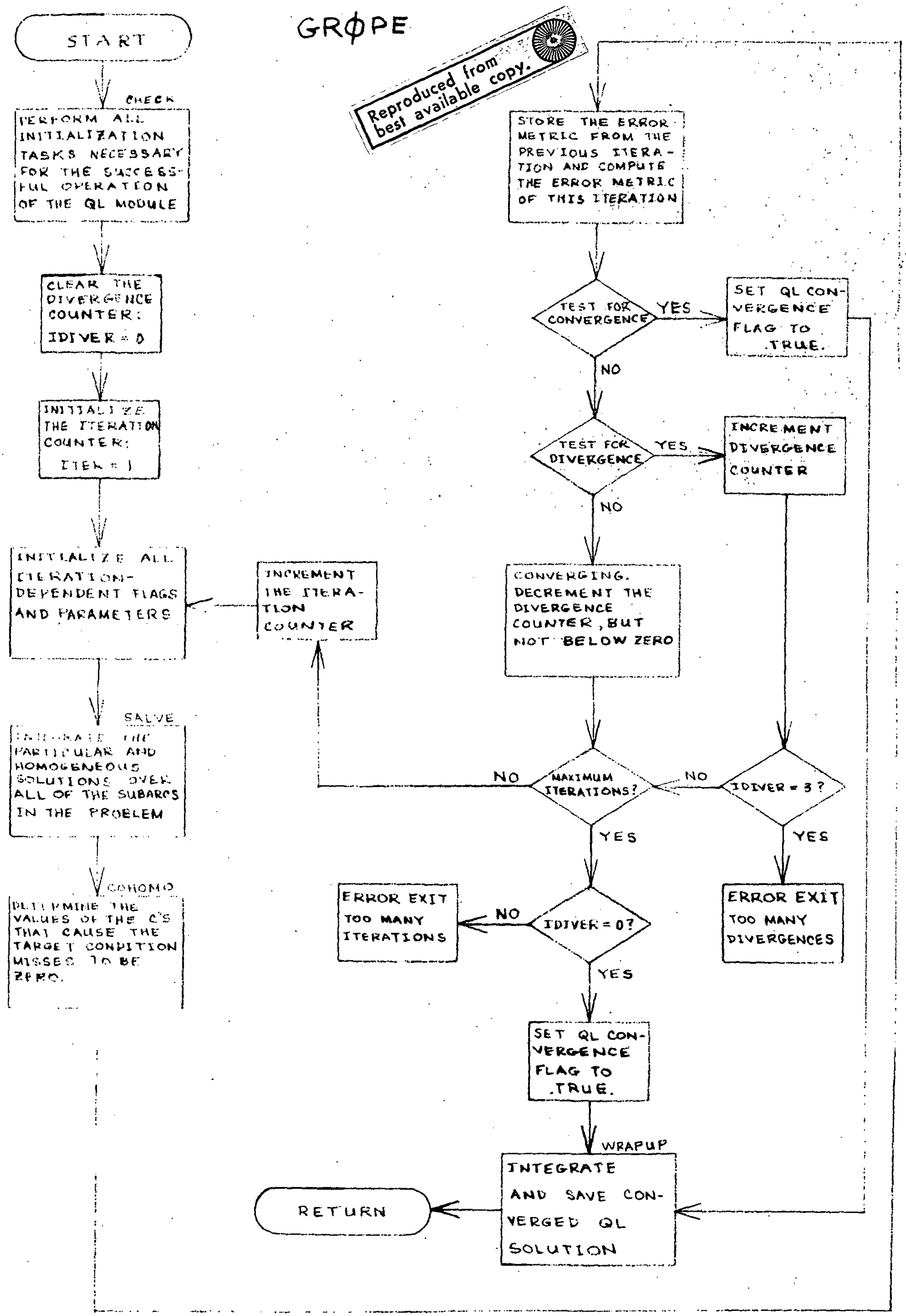
Step 1	CHECK
Step 2A	SALVE
Step 2B	COHOMO
Step 3	WRAPUP

The flow chart of subroutine GROPE on the next page depicts the sequence in which these steps are executed. The listing of GROPE following the flow chart is a CDC-compatible version. As a result, the transfers to the above

entry points appear as calls to OVERLAY. The correspondence of entry points to OVERLAY calls is:

CHECK	CALL OVERLAY (6HH75021, 4, 1, 6HRECALL)
SALVE	CALL OVERLAY (6HH75022, 4, 2, 6HRECALL)
COHOMO	CALL OVERLAY (6HH75023, 4, 3, 6HRECALL)
WRAPUP	CALL OVERLAY (6HH75024, 4, 4, 6HRECALL)

GRØPE



START
CHECK
PERFORM ALL INITIALIZATION TASKS NECESSARY FOR THE SUCCESSFUL OPERATION OF THE QL MODULE
CLEAR THE DIVERGENCE COUNTER: IDIVER = 0
INITIALIZE THE ITERATION COUNTER: ITER = 1
INITIALIZE ALL ITERATION-DEPENDENT FLAGS AND PARAMETERS
SALVE
INTEGRATE THE PARTICULAR AND HOMOGENEOUS SOLUTIONS OVER ALL OF THE SUBARCS IN THE PROBLEM
COHOMO
DETERMINE THE VALUES OF THE C'S THAT CAUSE THE TARGET CONDITION MISSES TO BE ZERO.
STORE THE ERROR METRIC FROM THE PREVIOUS ITERATION AND COMPUTE THE ERROR METRIC OF THIS ITERATION
TEST FOR CONVERGENCE
YES
SET QL CONVERGENCE FLAG TO .TRUE.
NO
TEST FOR DIVERGENCE
YES
INCREMENT DIVERGENCE COUNTER
NO
CONVERGING. DECREMENT THE DIVERGENCE COUNTER, BUT NOT BELOW ZERO
INCREMENT THE ITERATION COUNTER
NO
MAXIMUM ITERATIONS?
NO
IDIVER = 3?
YES
YES
ERROR EXIT TOO MANY ITERATIONS
NO
IDIVER = 0?
ERROR EXIT TOO MANY DIVERGENCES
YES
SET QL CONVERGENCE FLAG TO .TRUE.
WRAPUP
INTEGRATE AND SAVE CONVERGED QL SOLUTION
RETURN

760


```
      PROGRAM GROPE                                                     GROPE
C                                                                       GROPE
C                      THIS ROUTINE CONTROLS THE OVERALL FLOW OF THE QL GROPE
C                      MODULE OF PADS                                   GROPE
C                                                                       GROPE
      REAL MAG                                                          GROPE
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      COMMON /Z/ Z(50)                                                  Z
      COMMON /ZD/ ZD(50)                                                ZD
      COMMON /Y/ Y(820)                                                 Y
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      DIMENSION CDIFF(40), PC(9)                                        GROPE
      EQUIVALENCE (PC, PC1)                                             GROPE
      DATA GROPEX /6HGROPE /                                            GROPE
      DATA PC/6HGROPE , 18, 0, 8, 3*0, 40, 5/                           GROPE
C                      GET READY TO ITERATE                             GROPE
      CALL OVERLAY(6HH75021,4, 1, 6HRECALL)                             GROPE
C                      INITIALIZE QL DIVERGENCE COUNTER                 GROPE
      IDIVER = 0                                                        GROPE
      DO 111 ITER = 1, ITRMAX                                           GROPE
      CALL ETIME                                                        GROPE
      KPAGE = -1                                                        GROPE
C                      SET POINTERS TO SCRATCH FILES.  ITAPA HAS THE PAR- GROPE
C                      TICULAR AND HOMOGENEOUS SOLUTIONS FROM THE LAST  GROPE
C                      ITERATION.  ITAPB WILL CONTAIN THE SAME INFO. FROM GROPE
C                      THIS ITERATION.                                  GROPE
      IF(MOD(ITER, 2) .NE. 1) GO TO 101                                 GROPE   101
      ITAPA = 12                                                        GROPE
      ITAPB = 4                                                         GROPE
      GO TO 102                                                         GROPE   102
  101 ITAPA = 4                                                         GROPE
      ITAPB = 12                                                        GROPE
  102 REWIND 4                                                          GROPE
      REWIND 12                                                         GROPE
      DO 103 I = 1, NUP                                                 GROPE
  103 Y(I) = 0.                                                         GROPE
      MOM = 0                                                           GROPE
C                      SET KODES TO ZERO SO THAT NLDRV WILL ZERO OUT    GROPE
C                      JACOB                                            GROPE
      KODES = 0                                                         GROPE
C                      INTEGRATE PARTICULAR AND HOMOGENEOS SOLUTIONS OVER GROPE
C                      ALL THE SUBARCS.                                 GROPE
      CALL OVERLAY(6HH75022,4, 2, 6HRECALL)                             GROPE
C                      DETERMINE THE VALUES OF THE C*S THAT CAUSE THE   GROPE
C                      TARGET CONDITION MISSES TO BE ZERO.              GROPE
      CALL OVERLAY(6HH75023,4, 3, 6HRECALL)                             GROPE
C                      STORE ERROR METRIC FROM LAST ITERATION           GROPE
      RHOP = RHOC                                                       GROPE
C                      COMPUTE ERROR METRIC FOR THIS ITERATION          GROPE
      RHOC = MAG(C, NOKNOW)                                             GROPE
C                      PRINT ERROR METRIC                               GROPE
      WRITE(6, 1) RHOC                                                  GROPE
```

```
      1 FORMAT(1HO, 13HERROR METRIC=, E14.7)                            GROPE
C                      ARE WE CONVERGED                                 GROPE
        IF(RHOC - EPSLON) 108, 108, 109                                 GROPE
C                      YES.  SET QL CONVERGENCE FLAG                    GROPE
    108 KONVER = .TRUE.                                                 GROPE
        GO TO 112                                                       GROPE
C                      DID WE DIVERGE                                   GROPE
    109 IF(RHOC .LT. RHOP) GO TO 110                                    GROPE
C                      YES.  INCREMENT DIVERGENCE FLAG                  GROPE
        IDIVER = IDIVER + 1                                             GROPE
C                      HAVE WE ACCUMULATED 3 DIVERGENCES                GROPE
        IF(IDIVER .EQ. 3) CALL ERROR(GROPEX, -5, 0)                     GROPE
        GO TO 111                                                       GROPE
C                      CONVERGING.  DECREMENT DIVERGENCE FLAG           GROPE
    110 IDIVER = MAXO(0, IDIVER - 1)                                    GROPE
    111 CONTINUE                                                        GROPE
C                      RUN OUT OF ITERATIONS.  SHALL WE CALL IT CONVERGED GROPE
C                      ANYWAY                                           GROPE
        IF(IDIVER .EQ. 0) GO TO 1111                                    GROPE
C                      NO.  LET STEEP. DESC. GENERATE A BETTER STARTING GROPE
C                      SOLUTION IF IT WANTS TO.                         GROPE
        CALL ERROR(GROPEX, -3, 0)                                       GROPE
   1111 KONVER = .TRUE.                                                 GROPE
C                      CONVERGED.  SAVE CONVERGED SOLUTION AND PRINT IT GROPE
C                      OUT IN ALL ITS GLORY.                            GROPE
    112 CALL OVERLAY(6HH75024,4, 4, 6HRECALL)                           GROPE
        RETURN                                                          GROPE
        END                                                             GROPE
```

6 OCT 72 6.01-44

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D /(	11)	BNDRY BRANPT GROPE INTRPT NEWCS NLDRV NOMNAL WRAPUP	I I I I M I I I	C C C C C C C C
EPSLON	ϵ	I	QL iteration convergence criterion.	/GLOBAL/(	8)	CHECK GROPE	M I	EPSLON EPSLON
ITAPA		O	Number of the logical unit containing the quasitime histories of the particular and homogeneous solutions from the preceding QL iteration.	/CNTRL /(	3)	GROPE NOMNAL	O I	ITAPA ITAPA
ITAPB		O	Number of the logical unit onto which the quasitime histories of the particular and homogeneous solutions from the current QL iteration are written.	/CNTRL /(	4)	GROPE SALVE	O I	ITAPB ITAPB
ITER		M	QL iteration number.	/CNTRL /(	2)	ETIME GROPE OUTPUT	M M I	ITER ITER ITER
ITRMAX		I	Maximum number of QL iterations.	/GLOBAL/(	10)	CHECK GROPE	M I	ITRMAX ITRMAX
KODES		O	Not used.	/CNTRL /(	56)	GROPE NLDRV WRAPUP	O M O	KODES KODES KODES
KONVER		O	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON APPLY ARCIN COHOMO GROPE NLDRV OUTPUT RKUTT1	I I I O O I I I	KONVER KONVER KONVER KONVER KONVER KONVER KONVER KONVER
KPAGE		O	Not used.	/CNTRL /(	21)	CHECK GROPE	O O	KPAGE KPAGE
MOM		O	The number of homogeneous solutions currently being integrated.	/CNTRL /(	9)	GROPE INARC LINDRV NOMNAL SALVE WRAPUP	O M I I M M	MOM MOM MOM MOM MOM MOM
NOKNOW		I	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK COHOMO COSTAB COSTAI COSTAO GROPE MAGIC	I I M M M I I	NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW
NUP		I	Same as NU.	/CNTRL /(	23)	CHECK GROPE INARC	O I I	NUP NUP NUP
PC		D	Not used	/PC /(	1)	GROPE	O	PC
RHOC		M	The magnitude of the error in the current QL iteration.	/CNTRL /(	17)	CHECK GROPE	O M	RHOC RHOC
RHOP		M	The magnitude of the error in the preceding QL iteration.	/CNTRL /(	18)	CHECK GROPE	O M	RHOP RHOP
Y		O	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y /(	1)	GROPE INARC MADAMS QLTOSZ RKUTT1 SALVE WRAPUP	O M M I M M I	Y Y Y Y Y Y Y

763

BLØCK
ARCDAT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SREF	S_{ref}	Aerodynamic reference area (FT²)	/ARCDAT/(	1)	ARCIN BNDRY CHECK FETCH SALVE STATEF UT WRAPUP	I I I I I I I I	SREF ARCDA ARCDA ARCDA ARCDA SREF SREF ARCDA
EJ	A_{exit}	Nozzle exit area (FT²)	/ARCDAT/(	2)	ARCIN	I	EJ
XISP	I_{sp}	Vacuum specific impulse (SEC)	/ARCDAT/(	3)	ARCIN IMPULS	I I	XISP XISP
TMULT	T_{mult}	Thrust multiplier or number of engines	/ARCDAT/(	4)	ARCIN STATEF	I I	TMULT TMULT
DTNC	$\Delta\tau$	Integration interval (SEC)	/ARCDAT/(	5)	WRAPUP	I	DTNC
DTPI		Print frequency for trajectory	/ARCDAT/(	6)	WRAPUP	I	DTPI
IATM		Atmosphere option flag	/ARCDAT/(	7)	ARCIN NLDRV OUTPUT STATEF	I I I I	IATM IATM IATM IATM
IMOD		Control mode option flag	/ARCDAT/(	8)	ARCIN CONTRL NPLANE	I I I	IMODE IMODE IMODE
JAER		Aerodynamic model option flag	/ARCDAT/(	9)	AEROCO ARCIN OUTPUT STATEF UT	I I I I I	JAER JAER JAER JAER JAER
JPRO		Propulsion model option flag	/ARCDAT/(	10)	ARCIN IMPULS	I I	JPRO JPRO
QMAX	q_{MAX}	Maximum dynamic pressure instantaneous inequality limit (PSF)	/ARCDAT/(	11)			
GMAX	G_{MAX}	Maximum total acceleration g load	/ARCDAT/(	12)	AL5 NPLANE THROTL TH3	I I I I	GMAX GMAX GMAX GMAX
XLAMX	L_{MAX}	Maximum aerodynamic lift (LBS)	/ARCDAT/(	13)	NPLANE	I	XLMAX
HDMAX	$\dot{Q}_{MAX}$	Maximum heating rate inequality constraint	/ARCDAT/(	14)			
GMDOT	$\dot{\gamma}^*$	Pitch rate (DEG/SEC)	/ARCDAT/(	15)	ARCIN	I	GMDOT
ALFMAX	α_{MAX}	Maximum angle of attack (DEG)	/ARCDAT/(	16)	ARCIN NPLANE	I I	ALFMAX ALFMAX
PHMAX		Belly down flag	/ARCDAT/(	17)	CONTRL	I	PHMAX
MAEA		Curve number	/ARCDAT/(	18)	ARCIN STATEF	I I	MAEA MAEA
MAEB		Curve number	/ARCDAT/(	19)	STATEF	I	MAEB
MAEC		Curve number	/ARCDAT/(	20)	STATEF	I	MAEC
MAED		Curve number	/ARCDAT/(	21)	STATEF	I	MAED
MAEE		Curve number	/ARCDAT/(	22)	STATEF	I	MAEE
MAEF		Curve number	/ARCDAT/(	23)	STATEF	I	MAEF
MAEG		Curve number	/ARCDAT/(	24)	STATEF	I	MAEG
MT		Curve number -thrust table	/ARCDAT/(	25)	ARCIN	I	MT
MISP		Curve number XISP loss table	/ARCDAT/(	26)	ARCIN IMPULS	I I	MISP MISP
MAXCG		Curve number -xcg table	/ARCDAT/(	27)	STATEF	I	MXCG
MZCG		Curve number- zcg table	/ARCDAT/(	28)	STATEF	I	MZCG

765

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
MWDB		Description not input		/ARCDAT/(	30)	ARCIN	I	MWDB
MDB		Curve number - base drag table		/ARCDAT/(	31)	ARCIN STATEF	I I	MDB MDB
XCGR	X_{CGR}	Reference xcg location	(FT)	/ARCDAT/(	32)	UT	I	XCGR
ZCGR	Z_{CGR}	Reference zcg location	(FT)	/ARCDAT/(	33)	UT	I	ZCGR
XE	X_E	Engine thrust centroid body x station		/ARCDAT/(	34)	DL2	I	XE
ZE	Z_E	Engine thrust centroid body z station		/ARCDAT/(	35)	DL2 UT	I I	ZE ZE
XT	X_T	Aerodynamic trim surface body x station		/ARCDAT/(	36)	UT	I	XT
DREF	D_{ref}	Aerodynamic reference length		/ARCDAT/(	37)	STATEF UT	I I	DREF DREF
RHOB	ρ_b	Atmosphere base density for heating calculation	(LB/FT**3)	/ARCDAT/(	39)	NLDRV PDBCOL	I I	RHOB RHOB
QMULT	=0 OR 1	Heating flag multiplier		/ARCDAT/(	40)	NLDRV PDBCOL	I I	QMULT QMULT
REMAX	R_{eYMAX}	Maximum unit reynolds number inequality constraint		/ARCDAT/(	41)			
FRATE		Input rated vacuum thrust per engine	(LBS)	/ARCDAT/(	42)	ARCIN	M	FRATE

766

BLØCK
AXLE

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AV	a^v	The first word of a four word array that contains the vector of applied accelerations defined by Equation 2.4-2 in Vol.I of this document.	/AXLE	/(1)	APPLY APPLY NLDRV	M O I	AV AXLE AV
AG		Description not input	/AXLE	/(2)	APPLY NLDRV	M I	AG AG
AP		Description not input	/AXLE	/(3)	APPLY NLDRV	M I	AP AP
AM		Description not input	/AXLE	/(4)	APPLY NLDRV	O I	AM AM
AVV	$\partial a^v/\partial v$	The first entry in a 4x8 matrix that contains the total partial derivatives of the a vector with respect to the QL state vector (excluding the heating state), $\partial a/\partial y$.	/AXLE	/(5)	APPLY NLDRV	M I	AVV AVV
AGV		Description not input	/AXLE	/(6)	APPLY NLDRV	O I	AGV AGV
APV		Description not input	/AXLE	/(7)	APPLY NLDRV	O I	APV APV
AMV		Description not input	/AXLE	/(8)	APPLY NLDRV	O I	AMV AMV
AVG		Description not input	/AXLE	/(9)	NLDRV	I	AVG
AGG		Description not input	/AXLE	/(10)	APPLY NLDRV	M I	AGG AGG
APG		Description not input	/AXLE	/(11)	APPLY NLDRV	M I	APG APG
AMG		Description not input	/AXLE	/(12)	NLDRV	I	AMG
AVP		Description not input	/AXLE	/(13)	NLDRV	I	AVP
AMP		Description not input	/AXLE	/(16)	NLDRV	I	AMP
AVR		Description not input	/AXLE	/(17)	APPLY NLDRV	O I	AVR AVR
AGR		Description not input	/AXLE	/(18)	APPLY NLDRV	O I	AGR AGR
APR		Description not input	/AXLE	/(19)	APPLY NLDRV	O I	APR APR
AMR		Description not input	/AXLE	/(20)	APPLY NLDRV	O I	AMR AMR
AVO		Description not input	/AXLE	/(21)	NLDRV	I	AVO
AMO		Description not input	/AXLE	/(24)	NLDRV	I	AMO
AVM		Description not input	/AXLE	/(29)	APPLY NLDRV	O I	AVM AVM
AGM		Description not input	/AXLE	/(30)	APPLY NLDRV	O I	AGM AGM
APM		Description not input	/AXLE	/(31)	APPLY NLDRV	O I	APM APM
AMM		Description not input	/AXLE	/(32)	NLDRV	I	AMM
AVZ		Description not input	/AXLE	/(33)	NLDRV	I	AVZ
AGZ		Description not input	/AXLE	/(34)	NLDRV	I	AGZ
APZ		Description not input	/AXLE	/(35)	NLDRV	I	APZ
AMZ		Description not input	/AXLE	/(36)	NLDRV	I	AMZ
EAVV	$\delta a^v/\delta v$	The first entry in a 4x8 matrix that contains, when α is optimal, $\delta a/\delta y = \partial a/\partial y\|_\alpha = \text{constant}$ When α is nonoptimal, $\delta a/\delta y = \partial a/\partial y$	/AXLE	/(37)	APPLY NLDRV	I I	EAVV EAVV
EAGV		Description not input	/AXLE	/(38)	NLDRV	I	EAGV
EAPV		Description not input	/AXLE	/(39)	NLDRV	I	EAPV
EAMV		Description not input	/AXLE	/(40)	NLDRV	I	EAMV
EAVR		Description not input	/AXLE	/(49)	NLDRV	I	EAVR

768

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
EAGR		Description not input	/AXLE /(	50)	NLDRV	I	EAGR
EAPR		Description not input	/AXLE /(	51)	NLDRV	I	EAPR
EAMR		Description not input	/AXLE /(	52)	NLDRV	I	EAMR
EAVM		Description not input	/AXLE /(	61)	NLDRV	I	EAVM
EAGM		Description not input	/AXLE /(	62)	NLDRV	I	EAGM
EAPM		Description not input	/AXLE /(	63)	NLDRV	I	EAPM
EAMM		Description not input	/AXLE /(	64)	NLDRV	I	EAMM
EAVZ		Description not input	/AXLE /(	65)	NLDRV	I	EAVZ
EAGZ		Description not input	/AXLE /(	66)	NLDRV	I	EAGZ
EAPZ		Description not input	/AXLE /(	67)	NLDRV	I	EAPZ
EAMZ		Description not input	/AXLE /(	68)	NLDRV	I	EAMZ
AVVV	$\partial(\delta a^{v}/\delta V)/\partial V$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial V$.	/AXLE /(	69)	APPLY APPLY NLDRV	O I I	AVVV AVY AVVV
AGVV		Description not input	/AXLE /(	70)	APPLY NLDRV	O I	AGVV AGVV
APVV		Description not input	/AXLE /(	71)	APPLY NLDRV	O I	APVV APVV
AMVV		Description not input	/AXLE /(	72)	APPLY NLDRV	O I	AMVV AMVV
AVRV		Description not input	/AXLE /(	81)	APPLY NLDRV	M I	AVRV AVRV
AGRV		Description not input	/AXLE /(	82)	APPLY NLDRV	M I	AGRV AGRV
APRV		Description not input	/AXLE /(	83)	APPLY NLDRV	M I	APRV APRV
AMRV		Description not input	/AXLE /(	84)	APPLY NLDRV	O I	AMRV AMRV
AVMV		Description not input	/AXLE /(	93)	APPLY NLDRV	M I	AVMV AVMV
AGMV		Description not input	/AXLE /(	94)	APPLY NLDRV	M I	AGMV AGMV
APMV		Description not input	/AXLE /(	95)	APPLY NLDRV	M I	APMV APMV
AMMV		Description not input	/AXLE /(	96)	NLDRV	I	AMMV
AVZV		Description not input	/AXLE /(	97)	NLDRV	I	AVZV
AGZV		Description not input	/AXLE /(	98)	NLDRV	I	AGZV
APZV		Description not input	/AXLE /(	99)	NLDRV	I	APZV
AMZV		Description not input	/AXLE /(	100)	NLDRV	I	AMZV
AVVG	$\partial(\delta a^{v}/\delta V)/\partial V$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial Y$	/AXLE /(	101)	NLDRV	I	AVVG
AGVG		Description not input	/AXLE /(	102)	APPLY NLDRV	O I	AGVG AGVG
APVG		Description not input	/AXLE /(	103)	APPLY NLDRV	O I	APVG APVG
AMVG		Description not input	/AXLE /(	104)	NLDRV	I	AMVG
AVGG		Description not input	/AXLE /(	105)	NLDRV	I	AVGG
AMGG		Description not input	/AXLE /(	108)	NLDRV	I	AMGG
AVRG		Description not input	/AXLE /(	113)	NLDRV	I	AVRG
AGRG		Description not input	/AXLE /(	114)	APPLY NLDRV	O I	AGRG AGRG
APRG		Description not input	/AXLE /(	115)	APPLY NLDRV	O I	APRG APRG
AMRG		Description not input	/AXLE /(	116)	NLDRV	I	AMRG
AVMG		Description not input	/AXLE /(	125)	NLDRV	I	AVMG

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AGMG		Description not input	/AXLE	/(126)	APPLY NLDRV	O I	AGMG AGMG
APMG		Description not input	/AXLE	/(127)	APPLY NLDRV	O I	APMG APMG
AMMG		Description not input	/AXLE	/(128)	NLDRV	I	AMMG
AVZG		Description not input	/AXLE	/(129)	NLDRV	I	AVZG
AGZG		Description not input	/AXLE	/(130)	NLDRV	I	AGZG
APZG		Description not input	/AXLE	/(131)	NLDRV	I	APZG
AMZG		Description not input	/AXLE	/(132)	NLDRV	I	AMZG
AVVP	$\partial(\delta a^V/\delta V)/\partial\psi$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\psi$	/AXLE	/(133)	NLDRV	I	AVVP
AMVP		Description not input	/AXLE	/(136)	NLDRV	I	AMVP
AVGP		Description not input	/AXLE	/(137)	NLDRV	I	AVGP
AMGP		Description not input	/AXLE	/(140)	NLDRV	I	AMGP
AVPP		Description not input	/AXLE	/(141)	NLDRV	I	AVPP
AMPP		Description not input	/AXLE	/(144)	NLDRV	I	AMPP
AVVR	$\partial(\delta a^V/\delta V)/\partial h$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial h$	/AXLE	/(165)	APPLY NLDRV	O I	AVVR AVVR
AGVR		Description not input	/AXLE	/(166)	APPLY NLDRV	O I	AGVR AGVR
APVR		Description not input	/AXLE	/(167)	APPLY NLDRV	O I	APVR APVR
AMVR		Description not input	/AXLE	/(168)	APPLY NLDRV	M I	AMVR AMVR
AVGR		Description not input	/AXLE	/(169)	NLDRV	I	AVGR
AMGR		Description not input	/AXLE	/(172)	NLDRV	I	AMGR
AVPR		Description not input	/AXLE	/(173)	NLDRV	I	AVPR
AMPR		Description not input	/AXLE	/(176)	NLDRV	I	AMPR
AVRR		Description not input	/AXLE	/(177)	APPLY NLDRV	O I	AVRR AVRR
AGRR		Description not input	/AXLE	/(178)	APPLY NLDRV	O I	AGRR AGRR
APRR		Description not input	/AXLE	/(179)	APPLY NLDRV	O I	APRR APRR
AMRR		Description not input	/AXLE	/(180)	APPLY NLDRV	O I	AMRR AMRR
AVMR		Description not input	/AXLE	/(189)	APPLY NLDRV	M I	AVMR AVMR
AGMR		Description not input	/AXLE	/(190)	APPLY NLDRV	M I	AGMR AGMR
APMR		Description not input	/AXLE	/(191)	APPLY NLDRV	M I	APMR APMR
AMMR		Description not input	/AXLE	/(192)	NLDRV	I	AMMR
AVZR		Description not input	/AXLE	/(193)	NLDRV	I	AVZR
AGZR		Description not input	/AXLE	/(194)	NLDRV	I	AGZR
APZR		Description not input	/AXLE	/(195)	NLDRV	I	APZR
AMZR		Description not input	/AXLE	/(196)	NLDRV	I	AMZR
AVVO		Description not input	/AXLE	/(197)	NLDRV	I	AVVO
AMVO		Description not input	/AXLE	/(200)	NLDRV	I	AMVO
AVGO		Description not input	/AXLE	/(201)	NLDRV	I	AVGO
AMGO		Description not input	/AXLE	/(204)	NLDRV	I	AMGO
AVPO		Description not input	/AXLE	/(205)	NLDRV	I	AVPO
AVVO	$\partial(\delta a^V/\delta V)/\partial\rho$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\rho$	/AXLE	/(207)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AMPO		Description not input	/AXLE /(	208)	NLDRV	I	AMPO
AVRO		Description not input	/AXLE /(	209)	NLDRV	I	AVRO
AMRO		Description not input	/AXLE /(	212)	NLDRV	I	AMRO
AVOO		Description not input	/AXLE /(	213)	NLDRV	I	AVOO
AMOO		Description not input	/AXLE /(	216)	NLDRV	I	AMOO
AVVU	$\partial(\delta a^V/\delta V)/\partial\mu$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\mu$	/AXLE /(	239)			
AVVM		Description not input	/AXLE /(	261)	APPLY NLDRV	O I	AVVM AVVM
AGVM		Description not input	/AXLE /(	262)	APPLY NLDRV	O I	AGVM AGVM
APVM		Description not input	/AXLE /(	263)	APPLY NLDRV	O I	APVM APVM
AMVM		Description not input	/AXLE /(	264)	NLDRV	I	AMVM
AVGM		Description not input	/AXLE /(	265)	NLDRV	I	AVGM
AMGM		Description not input	/AXLE /(	268)	NLDRV	I	AMGM
AVPM		Description not input	/AXLE /(	269)	NLDRV	I	AVPM
AVVM	$\partial(\delta a^V/\delta V)/\partial m$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial m$	/AXLE /(	271)			
AMPM		Description not input	/AXLE /(	272)	NLDRV	I	AMPM
AVRM		Description not input	/AXLE /(	273)	APPLY NLDRV	O I	AVRM AVRM
AGRM		Description not input	/AXLE /(	274)	APPLY NLDRV	O I	AGRM AGRM
APRM		Description not input	/AXLE /(	275)	APPLY NLDRV	O I	APRM APRM
AMRM		Description not input	/AXLE /(	276)	NLDRV	I	AMRM
AVOM		Description not input	/AXLE /(	277)	NLDRV	I	AVOM
AMOM		Description not input	/AXLE /(	280)	NLDRV	I	AMOM
AVMM		Description not input	/AXLE /(	285)	APPLY NLDRV	O I	AVMM AVMM
AGMM		Description not input	/AXLE /(	286)	APPLY NLDRV	O I	AGMM AGMM
APMM		Description not input	/AXLE /(	287)	APPLY NLDRV	O I	APMM APMM
AMMM		Description not input	/AXLE /(	288)	NLDRV	I	AMMM
AVZM		Description not input	/AXLE /(	289)	NLDRV	I	AVZM
AGZM		Description not input	/AXLE /(	290)	NLDRV	I	AGZM
APZM		Description not input	/AXLE /(	291)	NLDRV	I	APZM
AMZM		Description not input	/AXLE /(	292)	NLDRV	I	AMZM
AVVZ		Description not input	/AXLE /(	293)	NLDRV	I	AVVZ
AGVZ		Description not input	/AXLE /(	294)	NLDRV	I	AGVZ
APVZ		Description not input	/AXLE /(	295)	NLDRV	I	APVZ
AMVZ		Description not input	/AXLE /(	296)	NLDRV	I	AMVZ
AVGZ		Description not input	/AXLE /(	297)	NLDRV	I	AVGZ
AMGZ		Description not input	/AXLE /(	300)	NLDRV	I	AMGZ
AVPZ		Description not input	/AXLE /(	301)	NLDRV	I	AVPZ
AVVZ	$\partial(\delta a^V/\delta V)/\partial\tau$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\tau$	/AXLE /(	303)			
AMPZ		Description not input	/AXLE /(	304)	NLDRV	I	AMPZ
AVRZ		Description not input	/AXLE /(	305)	NLDRV	I	AVRZ
AGRZ		Description not input	/AXLE /(	306)	NLDRV	I	AGRZ
APRZ		Description not input	/AXLE /(	307)	NLDRV	I	APRZ

771

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AMRZ		Description not input	/AXLE	/(368)	NLDRV	I	AMRZ
AVOZ		Description not input	/AXLE	/(309)	NLDRV	I	AVOZ
AMOZ		Description not input	/AXLE	/(312)	NLDRV	I	AMOZ
AVMZ		Description not input	/AXLE	/(317)	NLDRV	I	AVMZ
AGMZ		Description not input	/AXLE	/(318)	NLDRV	I	AGMZ
APMZ		Description not input	/AXLE	/(319)	NLDRV	I	APMZ
AMMZ		Description not input	/AXLE	/(320)	NLDRV	I	AMMZ
AVZZ		Description not input	/AXLE	/(321)	NLDRV	I	AVZZ
AGZZ		Description not input	/AXLE	/(322)	NLDRV	I	AGZZ
APZZ		Description not input	/AXLE	/(323)	NLDRV	I	APZZ
AMZZ		Description not input	/AXLE	/(324)	NLDRV	I	AMZZ
AVLV		Description not input	/AXLE	/(325)	APPLY NLDRV	I I	AVLV AVLV
AGLV		Description not input	/AXLE	/(326)	NLDRV	I	AGLV
APLV		Description not input	/AXLE	/(327)	NLDRV	I	APLV
AMLV		Description not input	/AXLE	/(328)	NLDRV	I	AMLV
AVLG		Description not input	/AXLE	/(329)	NLDRV	I	AVLG
AGLG		Description not input	/AXLE	/(330)	APPLY NLDRV	M I	AGLG AGLG
APLG		Description not input	/AXLE	/(331)	APPLY NLDRV	M I	APLG APLG
AMLG		Description not input	/AXLE	/(332)	NLDRV	I	AMLG
AVLP		Description not input	/AXLE	/(333)	NLDRV	I	AVLP
AGLP		Description not input	/AXLE	/(334)	APPLY NLDRV	M I	AGLP AGLP
AVLV	$\partial a^v/\partial\lambda_v$	The first entry in a 4x3 matrix that contains $\partial a/\partial\lambda_v$, $\partial a/\partial\lambda_y$ and $\partial a/\partial\lambda_\psi$	/AXLE	/(335)	APPLY NLDRV	M I	APLP APLP
AMLP		Description not input	/AXLE	/(336)	NLDRV	I	AMLP
AYLY		Description not input	/AXLE	/(337)	APPLY NLDRV	I I	AYLY AVVLV
AGVLV		Description not input	/AXLE	/(338)	NLDRV	I	AGVLV
APVLV		Description not input	/AXLE	/(339)	NLDRV	I	APVLV
AMVLV		Description not input	/AXLE	/(340)	NLDRV	I	AMVLV
AVVLV	$\partial(\delta a^v/\delta V)/\partial\lambda_v$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\lambda$	/AXLE	/(347)			
AVRLV		Description not input	/AXLE	/(349)	NLDRV	I	AVRLV
AGRLV		Description not input	/AXLE	/(350)	NLDRV	I	AGRLV
APRLV		Description not input	/AXLE	/(351)	NLDRV	I	APRLV
AMRLV		Description not input	/AXLE	/(352)	NLDRV	I	AMRLV
AVMLV		Description not input	/AXLE	/(361)	NLDRV	I	AVMLV
AGMLV		Description not input	/AXLE	/(362)	NLDRV	I	AGMLV
APMLV		Description not input	/AXLE	/(363)	NLDRV	I	APMLV
AMMLV		Description not input	/AXLE	/(364)	NLDRV	I	AMMLV
AVZLV		Description not input	/AXLE	/(365)	NLDRV	I	AVZLV
AGZLV		Description not input	/AXLE	/(366)	NLDRV	I	AGZLV
APZLV		Description not input	/AXLE	/(367)	NLDRV	I	APZLV
AMZLV		Description not input	/AXLE	/(368)	NLDRV	I	AMZLV
AVVLG		Description not input	/AXLE	/(369)	NLDRV	I	AVVLG
AGVLG		Description not input	/AXLE	/(370)	APPLY NLDRV	D I	AGVLG AGVLG
APVLG		Description not input	/AXLE	/(371)	APPLY NLDRV	D I	APVLG APVLG

772

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AMVLG		Description not input	/AXLE	/(372)	NLDRV	I	AMVLG
AVVLG	$\partial(\delta a^v/\delta v)/\partial\lambda_\gamma$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\lambda_\gamma$	/AXLE	/(379)			
AVRLG		Description not input	/AXLE	/(381)	NLDRV	I	AVRLG
AGRLG		Description not input	/AXLE	/(382)	APPLY NLDRV	O I	AGRLG AGRLG
APRLG		Description not input	/AXLE	/(383)	APPLY NLDRV	O I	APRLG APRLG
AMRLG		Description not input	/AXLE	/(384)	NLDRV	I	AMRLG
AVMLG		Description not input	/AXLE	/(393)	NLDRV	I	AVMLG
AGMLG		Description not input	/AXLE	/(394)	APPLY NLDRV	O I	AGMLG AGMLG
APMLG		Description not input	/AXLE	/(395)	APPLY NLDRV	O I	APMLG APMLG
AMMLG		Description not input	/AXLE	/(396)	NLDRV	I	AMMLG
AVZLG		Description not input	/AXLE	/(397)	NLDRV	I	AVZLG
AGZLG		Description not input	/AXLE	/(398)	NLDRV	I	AGZLG
APZLG		Description not input	/AXLE	/(399)	NLDRV	I	APZLG
AMZLG		Description not input	/AXLE	/(400)	NLDRV	I	AMZLG
AVVLP		Description not input	/AXLE	/(401)	NLDRV	I	AVVLP
AGVLP		Description not input	/AXLE	/(402)	APPLY NLDRV	O I	AGVLP AGVLP
APVLP		Description not input	/AXLE	/(403)	APPLY NLDRV	O I	APVLP APVLP
AMVLP		Description not input	/AXLE	/(404)	NLDRV	I	AMVLP
AVVLP	$\partial(\delta a^v/\delta v)/\partial\lambda_\psi$	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\lambda_\psi$	/AXLE	/(411)			
AVRLP		Description not input	/AXLE	/(413)	NLDRV	I	AVRLP
AGRLP		Description not input	/AXLE	/(414)	APPLY NLDRV	O I	AGRLP AGRLP
APRLP		Description not input	/AXLE	/(415)	APPLY NLDRV	O I	APRLP APRLP
AMRLP		Description not input	/AXLE	/(416)	NLDRV	I	AMRLP
AVMLP		Description not input	/AXLE	/(425)	NLDRV	I	AVMLP
AGMLP		Description not input	/AXLE	/(426)	APPLY NLDRV	O I	AGMLP AGMLP
APMLP		Description not input	/AXLE	/(427)	APPLY NLDRV	O I	APMLP APMLP
AMMLP		Description not input	/AXLE	/(428)	NLDRV	I	AMMLP
AVZLP		Description not input	/AXLE	/(429)	NLDRV	I	AVZLP
AGZLP		Description not input	/AXLE	/(430)	NLDRV	I	AGZLP
APZLP		Description not input	/AXLE	/(431)	NLDRV	I	APZLP
AMZLP		Description not input	/AXLE	/(432)	NLDRV	I	AMZLP

BLØCK
BICUBE

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AMIN	x_0	The smallest value of the first independent variable of a bivariate table.	/BICUBE/(	1)	BLINE	I	AMIN
AMAX	x_N	The largest value of the first independent variable of a bivariate table.	/BICUBE/(	2)	BLINE	I	AMAX
IF		Last file in the grid in which interpolation occurred.	/BICUBE/(	3)	BLINE	M	IF
IFMAX	N	Total number of files in grid.	/BICUBE/(	4)	BLINE	I	IFMAX
RMIN	y_0	The smallest value of the second independent variable of a bivariate table.	/BICUBE/(	5)	BLINE	I	RMIN
RMAX	y_M	The largest value of the second independent variable of a bivariate table.	/BICUBE/(	6)	BLINE	I	RMAX
IR		Last rank in the grid in which interpolation occurred.	/BICUBE/(	7)	BLINE	M	IR
IRMAX		Total number of ranks in grid.	/BICUBE/(	8)	BLINE	I	IRMAX
IUNIT		Logical unit number on which bicubic spline coefficients are stored for this table.	/BICUBE/(	9)	BLINE	I	IUNIT
IRECT		Grid rectangle associated with IR and IF.	/BICUBE/(	10)	BLINE	M	IRECT
IREC		Logical record on IUNIT that contains spline coefficients for rectangle IRECT.	/BICUBE/(	11)	BLINE	M	IREC
C		A 32 word array containing the spline coefficients for the two bivariate functions at rectangle IRECT.	/BICUBE/(	12)	BLINE	O	C
					BLINE	I	CL00
CL10		Description not input	/BICUBE/(	13)	BLINE	I	CL10
CL20		Description not input	/BICUBE/(	14)	BLINE	I	CL20
CL30		Description not input	/BICUBE/(	15)	BLINE	I	CL30
CL01		Description not input	/BICUBE/(	16)	BLINE	I	CL01
CL11		Description not input	/BICUBE/(	17)	BLINE	I	CL11
CL21		Description not input	/BICUBE/(	18)	BLINE	I	CL21
CL31		Description not input	/BICUBE/(	19)	BLINE	I	CL31
CL02		Description not input	/BICUBE/(	20)	BLINE	I	CL02
CL12		Description not input	/BICUBE/(	21)	BLINE	I	CL12
CL22		Description not input	/BICUBE/(	22)	BLINE	I	CL22
CL32		Description not input	/BICUBE/(	23)	BLINE	I	CL32
CL03		Description not input	/BICUBE/(	24)	BLINE	I	CL03
CL13		Description not input	/BICUBE/(	25)	BLINE	I	CL13
CL23		Description not input	/BICUBE/(	26)	BLINE	I	CL23
CL33		Description not input	/BICUBE/(	27)	BLINE	I	CL33
CD00		Description not input	/BICUBE/(	28)	BLINE	I	CD00
CD10		Description not input	/BICUBE/(	29)	BLINE	I	CD10
CD20		Description not input	/BICUBE/(	30)	BLINE	I	CD20
CD30		Description not input	/BICUBE/(	31)	BLINE	I	CD30
CD01		Description not input	/BICUBE/(	32)	BLINE	I	CD01
CD11		Description not input	/BICUBE/(	33)	BLINE	I	CD11
CD21		Description not input	/BICUBE/(	34)	BLINE	I	CD21
CD31		Description not input	/BICUBE/(	35)	BLINE	I	CD31
CD02		Description not input	/BICUBE/(	36)	BLINE	I	CD02
CD12		Description not input	/BICUBE/(	37)	BLINE	I	CD12
CD22		Description not input	/BICUBE/(	38)	BLINE	I	CD22
CD32		Description not input	/BICUBE/(	39)	BLINE	I	CD32
CD03		Description not input	/BICUBE/(	40)	BLINE	I	CD03
CD13		Description not input	/BICUBE/(	41)	BLINE	I	CD13
CD23		Description not input	/BICUBE/(	42)	BLINE	I	CD23
CD33		Description not input	/BICUBE/(	43)	BLINE	I	CD33
T		A 160 word array containing logical record IREC.	/BICUBE/(	44)	BLINE	I	T

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ALFA		A 31 word array containing the mesh $x_0, x_1, \ldots, x_n$	/BICUBE/(	204)	BLINE	I	ALFA
MACH		A 31 word array containing the mesh $y_0, y_1, \ldots, y_n$	/BICUBE/(	235)	BLINE	I	MACH

776

BLØCK
BLØCK

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IIC		A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND BRANPT CHECK COSTAB COSTAI COSTAO INTRPT SALVE	M I I I I I I I	IIC IIC IIC IIC IIC IIC IIC IIC
IICT		A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK COSTAB COSTAI COSTAO MAGIC SALVE	I M M O O I	IICT IICT IICT IICT IICT IICT
ITC		A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT	O I I I I I I	ITC ITC ITC ITC ITC ITC ITC
JTAB		An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT MAGIC	M I I I I I I I	JTAB JTAB JTAB JTAB JTAB JTAB JTAB JTAB
ITCT		A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND BRANPT CHECK COSTAB COSTAI INTRPT MAGIC	O I I O O I O	ITCT ITCT ITCT ITCT ITCT ITCT ITCT
LTAB		An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND BRANPT COSTAB COSTAI INTRPT MAGIC	O I O O I M	LTAB LTAB LTAB LTAB LTAB LTAB
NOKNOW		The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK COHOMO COSTAB COSTAI COSTAO GROPE MAGIC	I I M M M I I	NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW
NOC		An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY BRANPT COSTAB COSTAI COSTAO INARC INTRPT SALVE WRAPUP	I I O O O I I I I	NOC NOC NOC NOC NOC NOC NOC NOC NOC
VALIC		A 10x20 array containing the desired values of all the fixed (known) QL state variables. The columns correspond to the subarc starting points, the rows, to QL state variables.	/BLOCK /(	862)	BCOND SALVE	M I	VALIC VALIC
VALTC		A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK /(	1062)	BCOND BRANPT CHECK ENDPT INTRPT	O I I I I	VALTC VALTC VALTC VALTC VALTC

7.78

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IPAV		Description not input	/BLOCK /	(1262)	BCOND	O	IPAV
					ENDPT	I	IPAV
					INTRPT	I	IPAV

BLØCK
CNTRL

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
NU		The largest number of quantities requiring numerical integration per QL iteration.	/CNTRL /(	1)	CHECK	O	NU
ITER		QL iteration number.	/CNTRL /(	2)	ETIME GROPE OUTPUT	M M I	ITER ITER ITER
ITAPA		Number of the logical unit containing the quasitime histories of the particular and homogeneous solutions from the preceding QL iteration.	/CNTRL /(	3)	GROPE NOMNAL	O I	ITAPA ITAPA
ITAPB		Number of the logical unit onto which the quasitime histories of the particular and homogeneous solutions from the current QL iteration are written.	/CNTRL /(	4)	GROPE SALVE	O I	ITAPB ITAPB
JMIN		Not used.	/CNTRL /(	5)			
JMAX		Not used.	/CNTRL /(	6)			
LINES		Not used.	/CNTRL /(	7)	CHECK	O	LINES
KPT		The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND BNDRY FORCES MAGIC RKUTT1 SALVE WRAPUP	O O I O I M M	KPT KPT KPT KPT KPT KPT KPT
MOM		The number of homogeneous solutions currently being integrated.	/CNTRL /(	9)	GROPE INARC LINDRV NOMNAL SALVE WRAPUP	O M I I M M	MOM MOM MOM MOM MOM MOM
KARD		The total number of homogeneous solutions eventually to be integrated.	/CNTRL /(	10)	CHECK	M	KARD
INDX		An array of four words that indicate to Adams-Moulton integration in what order the derivatives of the particular and homogeneous solutions are stored.	/CNTRL /(	11)	BCOND MADAMS SALVE	M M O	INDX INDX INDX
NEWNOM		A logical flag that indicates to the Runge-Kutta integration whether or not the system Jacobian needs to be reevaluated.	/CNTRL /(	15)	INTERP LINDRV RKUTT1 SALVE WRAPUP	O M O O O	NEWNOM NEWNOM NEWNOM NEWNOM NEWNOM
CNT016		Not used.	/CNTRL /(	16)			
RHOC		The magnitude of the error in the current QL iteration.	/CNTRL /(	17)	CHECK GROPE	O M	RHOC RHOC
RHOP		The magnitude of the error in the preceding QL iteration.	/CNTRL /(	18)	CHECK GROPE	O M	RHOP RHOP
NPTS		The total number of points in the subarc.	/CNTRL /(	19)	BCOND BNDRY FORCES INARC MAGIC SALVE WRAPUP	O O I M O M O	NPTS NPTS NPTS NPTS NPTS NPTS NPTS
MINES		Not used.	/CNTRL /(	20)	CHECK	O	MINES
KPAGE		Not used.	/CNTRL /(	21)	CHECK GROPE	O O	KPAGE KPAGE
NNP		Number of QL state and costate variables.(18)	/CNTRL /(	22)	CHECK INTERP	M I	NNP NNP
NUP		Same as NU.	/CNTRL /(	23)	CHECK GROPE INARC	O I I	NUP NUP NUP

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IARC	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
					BCOND	M	IARC
					BNDRY	M	IARC
					BRANPT	I	IARC
					CHECK	M	IARC
					COSTAB	I	IARC
					COSTAI	I	IARC
					ENDPT	I	IARC
					FORCES	I	IARC
					INARC	M	IARC
					INTRPT	I	IARC
					MAGIC	M	IARC
					MARCH	I	IARC
					QLTOSZ	I	IARC
					SALVE	M	IARC
					WRAPUP	M	IARC
TRSTR		Not used.	/CNTRL /(	25)	CHECK	I	TRSTR
					ETIME	M	TRSTR
IMAX		Not used.	/CNTRL /(	26)			
KTIME		Not used.	/CNTRL /(	27)			
KONVER		Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON	I	KONVER
					APPLY	I	KONVER
					ARCIN	I	KONVER
					COMOMO	O	KONVER
					GROPE	O	KONVER
					NLDRV	I	KONVER
					OUTPUT	I	KONVER
					RKUTT1	I	KONVER
NOPRNT		Not used.	/CNTRL /(	29)	OUTPUT	M	NOPRNT
					SALVE	O	NOPRNT
					TRAJIN	O	IPRNT
INBDRY		Not used.	/CNTRL /(	30)	CHECK	O	INBDRY
NUPAGE		Not used.	/CNTRL /(	31)			
IVARY		Twenty word array not used.	/CNTRL /(	32)			
NN		The number of quantities currently being numerically integrated.	/CNTRL /(	52)	BNDRY	M	NN
					INARC	M	NN
					MADAMS	I	NN
					MAGIC	M	NN
					NOMNAL	I	NN
					RKUTT1	I	NN
					RKUTT2	I	NN
					SALVE	M	NN
					WRAPUP	M	NN
NOVARY		Not used.	/CNTRL /(	53)			
PLAST		Not used.	/CNTRL /(	54)			
ZLAST		Not used.	/CNTRL /(	55)			
KODES		Not used.	/CNTRL /(	56)	GROPE	O	KODES
					NLDRV	M	KODES
					WRAPUP	O	KODES

782

BLØCK
D

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
X	x	The quasitime variable.	/D	/(1)	AL4	I	X
					BNDRY	O	X
					ERROR	I	X
					FETCH	O	X
					FORCES	I	X
					INARC	M	X
					INTERP	I	X
					MADAMS	M	X
					RKUTT1	M	X
					RKUTT2	M	X
					SALVE	M	X
					STATEF	I	X
					WRAPUP	M	TT
H	h	Integration step size in quasitime.	/D	/(2)	AL4	I	H
					INARC	M	H
					MADAMS	I	H
					RKUTT1	I	H
					RKUTT2	I	H
					SALVE	M	H
					WRAPUP	M	DT
XI		A four word array containing the first four values of quasitime in the subarc.	/D	/(3)	INTERP	I	XI
					SALVE	O	XI
MAGBV		The magnitude of all of the decired values of the state target conditions.	/D	/(7)	CHECK	M	MAGBV
					NEWCS	I	MAGBV
ERR		Convergence criterion of iteration for the c's.	/D	/(8)	CHECK	O	ERR
					NEWCS	I	ERR
D9		Not used.	/D	/(9)			
D10		Not used.	/D	/(10)			
C	c	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D	/(11)	BNDRY	I	C
					BRANPT	I	C
					GROPE	I	C
					INTRPT	I	C
					NEWCS	M	C
					NLDRV	I	C
					NOMNAL	I	C
					WRAPUP	I	C
CSAVE		A forty word array not used.	/D	/(51)			
V	V	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
					AL4	I	V
					AL7	I	V
					AL8	I	V
					AL9	I	V
					BCOND	I	NOM
					BNDRY	O	NOM
					BRANPT	M	NOM
					CONTRL	I	V
					ENDPT	I	NOM
					ENVPRO	I	V
					FETCH	O	NOM
					INTERP	M	V
					INTRPT	M	NOM
					NLDRV	O	NOM
					NLDRV	I	V
					OUTPUT	I	V
					PDBCOL	I	V
					STATEF	I	V
					WRAPUP	I	V
NOM		A twenty word array containing the state and costate vectors.	/D	/(91)			
GAM	γ	Relative flight path angle. (RAD)	/D	/(92)	ARCIN	I	GAM
					ENVPRO	I	GAM
					OUTPUT	I	GAM
					STATEF	I	GAM
					WRAPUP	I	GAM
PSI	ψ	Relative azimuth angle. (RAD)	/D	/(93)	OUTPUT	I	PSI
					STATEF	I	PSI
					WRAPUP	I	PSI

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALT	h	Altitude	(FT)	/D	/(94)	OUTPUT	I	ALT
						STATEF	I	ALT
						WRAPUP	I	ALT
RHO	ρ	Latitude	(RAD)	/D	/(95)	AL9	I	RHO
						OUTPUT	I	RHO
						STATEF	I	RHO
						WRAPUP	I	RHO
MU	μ	Relative longitude	(RAD)	/D	/(96)	OUTPUT	I	MU
						PDBCQL	I	MU
						WRAPUP	I	MU
M	m	Mass	(G'S)	/D	/(97)	AL4	I	M
						ALT	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
TAU	τ	Subarc duration	(SEC)	/D	/(98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU
HT	Q	Heating	(BTU)	/D	/(99)	OUTPUT	I	HT
						WRAPUP	I	HT
LV	λ_v	Relative velocity costate		/D	/(100)	AL1	I	LV
						CONTRL	I	LV
						NLDRV	I	LV
						NPLANE	I	LV
						OUTPUT	I	LV
						WRAPUP	I	LV
LGAM	λ_γ	Relative flight path angle costate		/D	/(101)	AL1	I	LGAM
						ARCIN	I	LGAM
						CONTRL	I	LGAM
						NLDRV	I	LGAM
						OUTPUT	I	LGAM
						WRAPUP	I	LGAM
LPSI	λ_ψ	Relative azimuth angle costate		/D	/(102)	AL1	I	LPSI
						ARCIN	I	LPSI
						CONTRL	I	LPSI
						NLDRV	I	LPSI
						OUTPUT	I	LPSI
						WRAPUP	I	LPSI
LR	λ_R	Altitude costate		/D	/(103)	NLDRV	I	LR
						OUTPUT	I	LR
						WRAPUP	I	LR
LRHO	λ_ρ	Latitude costate		/D	/(104)	NLDRV	I	LRHO
						OUTPUT	I	LRHO
						WRAPUP	I	LRHO
LMU	λ_μ	Relative longitude costate		/D	/(105)	NLDRV	I	LMU
						OUTPUT	I	LMU
						WRAPUP	I	LMU
LM	λ_m	Mass costate		/D	/(106)	NLDRV	I	LM
						OUTPUT	I	LM
						WRAPUP	I	LM
LTAU	λ_τ	Subarc duration costate		/D	/(107)	OUTPUT	I	LTAU
						WRAPUP	I	LTAU
LHT	λ_Q	Heating costate		/D	/(108)	NLDRV	I	LHT
						WRAPUP	I	LHT
D109		Not used.		/D	/(109)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
D110		Not used.	/D	/(110)			
BV		A forty word array not used.	/D	/(111)			
ZSAVE		A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D	/(151)	BCOND BRANPT COSTAB COSTAI INTRPT PDBCQL SALVE	O I I I I I I	ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE
QT		A twenty word array containing the values from the initial arc of the successive subarcs' durations	/D	/(171)	INARC WRAPUP	O M	QT QT
NPOINT		A twenty word array containing the number of points in each subarc.	/D	/(191)	INARC SALVE	O I	NPOINT NPOINT
DELT		A twenty word array containing the quasitime compute interval for each subarc.	/D	/(211)	CHECK ERROR INARC	O I I	DELT DELT DELT

BLØCK
DYNA

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XX		Fraction of subarc that has transpired	/DYNA	/(1)	ARCIN	O	XX
					ERROR	I	XX
					OUTPUT	I	XX
					STATEF	M	XX
TIME		Trajectory time (SEC)	/DYNA	/(2)	ENVPRO	I	TIME
					OUTPUT	I	TIME
					PDBCOL	I	TIME
					STATEF	M	TIME
					WRAPUP	I	TIME
SINGAM	$\sin\gamma$	See symbol	/DYNA	/(3)	AL1	I	SINGAM
					AL4	I	SINGAM
					AL7	I	SINGAM
					AL8	I	SINGAM
					AL9	I	SINGAM
					CONTRL	I	SINGAM
					NLDRV	I	SINGAM
					PDBCOL	I	SINGAM
					STATEF	M	SINGAM
COSGAM	$\cos\gamma$	See symbol	/DYNA	/(4)	AL1	I	COSGAM
					AL4	I	COSGAM
					AL7	I	COSGAM
					AL8	I	COSGAM
					AL9	I	COSGAM
					CONTRL	I	COSGAM
					NLDRV	I	COSGAM
					OUTPUT	I	COSGAM
					PDBCOL	I	COSGAM
					STATEF	M	COSGAM
OMEGA	ω	Earth rotation rate (RAD/SEC)	/DYNA	/(5)	AL4	I	OMEGA
					AL7	I	OMEGA
					CONTRL	I	OMEGA
					PDBCOL	I	OMEGA
					TRAJIN	M	OMEGA
OMEGA2	ω^2	See symbol	/DYNA	/(6)	AL4	I	OMEGA2
					AL7	I	OMEGA2
					AL8	I	OMEGA2
					AL9	I	OMEGA2
					NLDRV	I	OMEGA2
					TRAJIN	O	OMEGA2
R	R	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
					AL7	I	R
					AL8	I	R
					AL9	I	R
					CONTRL	I	R
					ENVPRO	I	R
					NLDRV	I	R
					PDBCOL	I	R
					QLTOSZ	I	R
					STATEF	M	R
G	g	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA	/(8)	AL4	I	G
					AL7	I	G
					AL8	I	G
					AL9	I	G
					CONTRL	I	G
					NLDRV	I	G
					STATEF	M	G
SINA	$\sin\alpha$	See symbol	/DYNA	/(9)	AL1	I	SINA
					AL4	I	SINA
					AL6	I	SINA
					AL7	I	SINA
					AL8	I	SINA
					AL9	I	SINA
					APPLY	I	SINA
					CONTRL	I	SINA
					OUTPUT	I	SINA
					TH3	I	SINA
					UT	M	SINA

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
COSA	$\cos\alpha$	See symbol		/DYNA /(	10)	AL1	I	COSA
						AL4	I	COSA
						AL6	I	COSA
						AL7	I	COSA
						AL8	I	COSA
						AL9	I	COSA
						APPLY	I	COSA
						CONTRL	I	COSA
						NLDRV	I	COSA
						OUTPUT	I	COSA
						TH3	I	COSA
						UT	M	COSA
DYN011		Not used.		/DYNA /(	11)			
OMEGAT	2ω	See symbol		/DYNA /(	12)	NLDRV	I	OMEGAT
						TRAJIN	O	OMEGAT
TAMP	T_a	Atmospheric temperature	(DEG-R)	/DYNA /(	13)	STATEF	I	TAMP
PA	P_a	Atmospheric pressure	(LBS/FT^2)	/DYNA /(	14)	IMPULS	I	PA
						NLDRV	I	PA
						OUTPUT	I	PA
						TH2	I	PA
RO	ρ_a	Atmospheric density	(SLGS/FT^3)	/DYNA /(	15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCQL	I	RO
						STATEF	I	RO
CS	a	Speed of sound	(FT/SEC)	/DYNA /(	16)	OUTPUT	I	CS
						STATEF	I	CS
TEMPR	$\partial T_a/\partial R$	See symbol		/DYNA /(	17)			
PAR	$\partial P_a/\partial R$	See symbol		/DYNA /(	18)	APPLY	I	PAR
						TH2	I	PAR
ROR	$\partial \rho_a/\partial R$	See symbol		/DYNA /(	19)	AL7	I	ROR
						AL8	I	ROR
						AL9	I	ROR
						NLDRV	I	ROR
						PDBCQL	I	ROR
						STATEF	I	ROR
CSR	$\partial a/\partial R$	See symbol		/DYNA /(	20)	STATEF	I	CSR
TEMPRR	$\partial^2 T_a/\partial R^2$	See symbol		/DYNA /(	21)			
PARR	$\partial^2 P_a/\partial R^2$	See symbol		/DYNA /(	22)	APPLY	I	PARR
						TH2	I	PARR
RORR	$\partial^2 \rho_a/\partial R^2$	See symbol		/DYNA /(	23)	AL7	I	RORR
						AL8	I	RORR
						AL9	I	RORR
						NLDRV	I	RORR
						STATEF	I	RORR
CSRR	$\partial^2 a/\partial R^2$	See symbol		/DYNA /(	24)	STATEF	I	CSRR
KODE		Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.		/DYNA /(	25)	APPLY	I	KODE
						ARCIN	O	KODE
						CONTRL	M	KODE
						FORCES	I	KODE
						NLDRV	I	KODE
						STATEF	I	KODE
MACH	M	Mach number		/DYNA /(	26)	AEROCO	I	MACH
						ENVPRO	I	MACH
						OUTPUT	I	MACH
						STATEF	M	MACH

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
Q	q	Dynamic pressure	(LBS/FT^2)	/DYNA	/(27)	ENVPRO	I	Q
						OUTPUT	I	Q
						PDBCOL	I	Q
						STATEF	M	Q
						UT	I	Q
QV	$\partial q/\partial V$	See symbol		/DYNA	/(28)	PDBCOL	I	QV
						STATEF	M	QV
						UT	I	QV
QR	$\partial q/\partial R$	See symbol		/DYNA	/(29)	PDBCOL	I	QR
						STATEF	M	QR
						UT	I	QR
QVV	$\partial^2 q/\partial V^2$	See symbol		/DYNA	/(30)	STATEF	M	QVV
						UT	I	QVV
QVR	$\partial^2 q/\partial V \partial R$	See symbol		/DYNA	/(31)	STATEF	M	QVR
						UT	I	QVR
QRR	$\partial^2 q/\partial R^2$	See symbol		/DYNA	/(32)	STATEF	M	QRR
						UT	I	QRR
FVAC		Total vacuum thrust (rocket)	(LBS)	/DYNA	/(33)	APPLY	I	FVAC
						ARCIN	M	FVAC
						IMPULS	M	FVAC
						NLORV	I	FVAC
						STATEF	M	FVAC
						TH2	I	FVAC
FVACV		Not used.		/DYNA	/(34)			
FVACR		Not used.		/DYNA	/(35)			
FVACM		Not used.		/DYNA	/(36)			
FVACT		Not used.		/DYNA	/(37)	ARCIN	I	FVACT
						STATEF	M	FVACT
						TH2	I	FVACT
FVACVV		Not used.		/DYNA	/(38)			
FVACVR		Not used.		/DYNA	/(39)			
FVACRR		Not used.		/DYNA	/(40)			
FVACTT		Not used.		/DYNA	/(41)	ARCIN	I	FVACTT
						STATEF	M	FVACTT
						TH2	I	FVACTT
T	T	Thrust	(LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
MACHV	$\partial M/\partial V$	See symbol		/DYNA	/(43)	STATEF	M	MACHV
						UT	I	MACHV
MACHR	$\partial M/\partial R$	See symbol		/DYNA	/(44)	STATEF	M	MACHR
						UT	I	MACHR
ISP	I_{SP}	Vacuum specific impulse	(SECS)	/DYNA	/(45)	APPLY	I	ISP
						ARCIN	O	ISP
						IMPULS	O	ISP
ISPV		Not used.		/DYNA	/(46)			
ISPR		Not used.		/DYNA	/(47)			
ISPM		Not used.		/DYNA	/(48)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ISPT		Not used.		/DYNA	/(49)			
ISPVV		Not used.		/DYNA	/(50)			
ISPVR		Not used.		/DYNA	/(51)			
ISPVM		Not used.		/DYNA	/(52)			
ISPVT		Not used.		/DYNA	/(53)			
ISPRR		Not used.		/DYNA	/(54)			
ISPRM		Not used.		/DYNA	/(55)			
ISPRT		Not used.		/DYNA	/(56)			
ISPMM		Not used.		/DYNA	/(57)			
ISPMT		Not used.		/DYNA	/(58)			
ISPTT		Not used.		/DYNA	/(59)			
LIFT	L	Aerodynamic lift	(LBS)	/DYNA	/(60)	AL4	I	LIFT
						AL5	I	LIFT
						AL6	I	LIFT
						APPLY	I	LIFT
						CONTRL	I	LIFT
						ENVPRO	I	LIFT
						OUTPUT	I	LIFT
						TH3	I	LIFT
						UT	O	LIFT
LIFTV	$\partial L/\partial V$	See symbol		/DYNA	/(61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTR	$\partial L/\partial R$	See symbol		/DYNA	/(62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTA	$\partial L/\partial \alpha$	See symbol		/DYNA	/(63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTVV	$\partial^2 L/\partial V^2$	See symbol		/DYNA	/(64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV
LIFTVR	$\partial^2 L/\partial V \partial R$	See symbol		/DYNA	/(65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	See symbol		/DYNA	/(66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTRR	$\partial^2 L/\partial R^2$	See symbol		/DYNA	/(67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBR	CODE	VAR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	See symbol		/DYNA /(	68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
DRAG	D	Aerodynamic drag	(LBS)	/DYNA /(	69)	AL5	I	DRAG
						AL7	I	DRAG
						AL8	I	DRAG
						AL9	I	DRAG
						APPLY	I	DRAG
						CONTRL	I	DRAG
						ENVPRQ	I	DRAG
						NLDRV	I	DRAG
						OUTPUT	I	DRAG
						TH3	I	DRAG
						UT	M	DRAG
DRAGV	$\partial D/\partial V$	See symbol		/DYNA /(	70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGR	$\partial D/\partial R$	See symbol		/DYNA /(	71)	AL5	I	DRAGR
						AL7	I	DRAGR
						AL8	I	DRAGR
						AL9	I	DRAGR
						APPLY	I	DRAGR
						TH3	I	DRAGR
						UT	M	DRAGR
DRAGA	$\partial D/\partial \alpha$	See symbol		/DYNA /(	72)	AL1	I	DRAGA
						AL5	I	DRAGA
						AL7	I	DRAGA
						AL8	I	DRAGA
						AL9	I	DRAGA
						APPLY	I	DRAGA
						TH3	I	DRAGA
						UT	M	DRAGA
DRAGVV	$\partial^2 D/\partial V^2$	See symbol		/DYNA /(	73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV
DRAGVR	$\partial^2 D/\partial V \partial R$	See symbol		/DYNA /(	74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	See symbol		/DYNA /(	75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
DRAGRR	$\partial^2 D/\partial R^2$	See symbol		/DYNA /(	76)	AL5	I	DRAGRR
						AL7	I	DRAGRR
						AL8	I	DRAGRR
						AL9	I	DRAGRR
						APPLY	I	DRAGRR
						TH3	I	DRAGRR
						UT	M	DRAGRR

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	See symbol	/DYNA /(	77)	AL1	I	DRAGRA
					AL5	I	DRAGRA
					AL7	I	DRAGRA
					AL8	I	DRAGRA
					AL9	I	DRAGRA
					APPLY	I	DRAGRA
					TH3	I	DRAGRA
					UT	M	DRAGRA
DRAGAA	$\partial^2 D/\partial \alpha^2$	See symbol	/DYNA /(	78)	AL1	I	DRAGAA
					AL5	I	DRAGAA
					AL7	I	DRAGAA
					AL8	I	DRAGAA
					AL9	I	DRAGAA
					APPLY	I	DRAGAA
					TH3	I	DRAGAA
					UT	M	DRAGAA
ALPHA	α	Angle of attack (RAD)	/DYNA /(	79)	AEROCO	I	ALPHA
					ALGCON	M	ALPHA
					AL2	I	ALPHA
					ARCIN	M	ALPHA
					CONTRL	M	ALPHA
					ENVPRO	I	ALPHA
					MOMECO	I	ALPHA
					NPLANE	M	ALPHA
					OUTPUT	I	ALPHA
					TRAJIN	O	ALPHA
					UT	I	ALPHA
					WRAPUP	I	ALPHA
PHI	ϕ	Bank angle (RAD)	/DYNA /(	80)	CONTRL	M	PHI
					OUTPUT	I	PHI
					WRAPUP	I	PHI
LIFTM	$\partial L/\partial m$	See symbol	/DYNA /(	81)	AL4	I	LIFTM
					AL5	I	LIFTM
					AL6	I	LIFTM
					APPLY	I	LIFTM
					TH3	I	LIFTM
					UT	O	LIFTM
LIFTVM	$\partial^2 L/\partial V \partial m$	See symbol	/DYNA /(	82)	AL4	I	LIFTVM
					AL5	I	LIFTVM
					AL6	I	LIFTVM
					APPLY	I	LIFTVM
					TH3	I	LIFTVM
					UT	O	LIFTVM
LIFTRM	$\partial^2 L/\partial R \partial m$	See symbol	/DYNA /(	83)	AL4	I	LIFTRM
					AL5	I	LIFTRM
					AL6	I	LIFTRM
					APPLY	I	LIFTRM
					TH3	I	LIFTRM
					UT	O	LIFTRM
LIFTMM	$\partial^2 L/\partial m^2$	See symbol	/DYNA /(	84)	AL4	I	LIFTMM
					AL5	I	LIFTMM
					AL6	I	LIFTMM
					APPLY	I	LIFTMM
					TH3	I	LIFTMM
					UT	O	LIFTMM
LIFTMA	$\partial^2 L/\partial m \partial \alpha$	See symbol	/DYNA /(	85)	AL1	I	LIFTMA
					AL4	I	LIFTMA
					AL5	I	LIFTMA
					AL6	I	LIFTMA
					APPLY	I	LIFTMA
					TH3	I	LIFTMA
					UT	O	LIFTMA

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DBR	$\partial D_b/\partial R$	See symbol		/DYNA /(	86)	AL1	I	DBR
						AL4	I	DBR
						AL6	I	DBR
						AL7	I	DBR
						AL8	I	DBR
						AL9	I	DBR
						APPLY	I	DBR
						STATEF	I	DBR
						TH3	I	DBR
						UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	See symbol		/DYNA /(	87)	AL4	I	DBRR
						AL6	I	DBRR
						AL7	I	DBRR
						AL8	I	DBRR
						AL9	I	DBRR
						APPLY	I	DBRR
						STATEF	I	DBRR
						TH3	I	DBRR
						UT	I	DBRR
GAMMAD		Pitch rate	(RAD/SEC)	/DYNA /(	88)	AL4	I	GAMMAD
						ARCIN	O	GAMMAD
						CONTRL	I	GAMMAD
						NLDRV	I	GAMMAD
AE	A_{exit}	Total nozzle exit area	(FT^2)	/DYNA /(	89)	APPLY	I	AE
						ARCIN	O	AE
						IMPULS	I	AE
						NLDRV	I	AE
						TH2	I	AE
TAX		Not used.		/DYNA /(	90)			
W	W	Weight	(LBS)	/DYNA /(	91)	AL5	I	W
						ENVPRO	I	W
						OUTPUT	I	W
						PDBCOL	I	W
						QLTOSZ	I	W
						STATEF	M	W
						TH3	I	W
SINPHI	$\sin\phi$	See symbol		/DYNA /(	92)	AL1	I	SINPHI
						AL4	I	SINPHI
						APPLY	I	SINPHI
						CONTRL	M	SINPHI
						OUTPUT	I	SINPHI
COSPHI	$\cos\phi$	See symbol		/DYNA /(	93)	AL1	I	COSPHI
						AL4	I	COSPHI
						APPLY	I	COSPHI
						ARCIN	O	COSPHI
						CONTRL	M	COSPHI
						OUTPUT	I	COSPHI
SINPSI	$\sin\psi$	See symbol		/DYNA /(	94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCOL	I	SINPSI
						STATEF	O	SINPSI
COSPSI	$\cos\psi$	See symbol		/DYNA /(	95)	AL4	I	COSPSI
						AL7	I	COSPSI
						AL8	I	COSPSI
						AL9	I	COSPSI
						CONTRL	I	COSPSI
						NLDRV	I	COSPSI
						PDBCOL	I	COSPSI
						STATEF	O	COSPSI

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SINRHO	$\sin\rho$	See symbol	/DYNA /(	96)	AL4 AL7 AL8 AL9 CONTRL NLDRV OUTPUT PDBCQL STATEF	I I I I I I I I M	SINRHO SINRHO SINRHO SINRHO SINRHO SINRHO SINRHO SINRHO SINRHO
COSRHO	$\cos\rho$	See symbol	/DYNA /(	97)	AL4 AL7 AL8 AL9 CONTRL NLDRV OUTPUT PDBCQL STATEF	I I I I I I I I M	COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO
SINROR		Not used.	/DYNA /(	98)			
COSROR		Not used.	/DYNA /(	99)			
MUR		Not used.	/DYNA /(	100)			
XKG	k_γ	Algebraic equation used in vertical rise and pitchover	/DYNA /(	101)	AL4 CONTRL	I M	XKG XKG
XKP	k_ψ	Algebraic equation used in vertical rise and pitchover	/DYNA /(	102)	AL4 CONTRL	I M	XKP XKP
AKIN		Not used.	/DYNA /(	103)			
CD0	C_{D_0}	Drag coefficient at $\alpha = 0$	/DYNA /(	104)	AEROCO STATEF	I I	CD0 CD0
CD0M	$\partial C_{D_0}/\partial M$	See symbol	/DYNA /(	105)	AEROCO STATEF	I I	CD0M CD0M
CL0	C_{L_0}	Lift coefficient at $\alpha = 0$	/DYNA /(	106)	AEROCO STATEF	I I	CL0 CL0
FK	k	Induced drag coefficient	/DYNA /(	107)	AEROCO STATEF	I I	FK FK
XCGM	$\partial X_{CG}/\partial m$	See symbol	/DYNA /(	108)	DL2 STATEF UT	I M I	XCGM XCGM XCGM
XCGMM	$\partial^2 X_{CG}/\partial m^2$	See symbol	/DYNA /(	109)	DL2 STATEF UT	I M I	XCGMM XCGMM XCGMM
ZCGM	$\partial Z_{CG}/\partial m$	See symbol	/DYNA /(	110)	DL2 STATEF UT	I M I	ZCGM ZCGM ZCGM
ZCGMM	$\partial^2 Z_{CG}/\partial m^2$	See symbol	/DYNA /(	111)	DL2 STATEF UT	I M I	ZCGMM ZCGMM ZCGMM
XJV	$\partial \dot{j}/\partial V$	See symbol	/DYNA /(	112)	DL2 STATEF UT	I O I	XJV XJV XJV
XJR	$\partial \dot{j}/\partial R$	See symbol	/DYNA /(	113)	DL2 STATEF UT	I O I	XJR XJR XJR
XJVV	$\partial^2 \dot{j}/\partial V^2$	See symbol	/DYNA /(	114)	DL2 STATEF UT	I O I	XJVV XJVV XJVV
XJVR	$\partial^2 \dot{j}/\partial V \partial R$	See symbol	/DYNA /(	115)	DL2 STATEF UT	I O I	XJVR XJVR XJVR
XJRR	$\partial^2 \dot{j}/\partial R^2$	See symbol	/DYNA /(	116)	DL2 STATEF UT	I O I	XJRR XJRR XJRR
MACHVR	$\partial^2 m/\partial V \partial R$	See symbol	/DYNA /(	117)	STATEF UT	O I	MACHVR MACHVR

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
MACHRR	$\partial^2 M/\partial R^2$	See symbol	/DYNA	/(118)	STATEF UT	M I	MACHRR MACHRR
SIN2RO	$\sin 2\rho$	See symbol	/DYNA	/(119)	AL4 AL7 AL8 NLORV STATEF	I I I I M	SIN2RO SIN2RO SIN2RO SIN2RO SIN2RO
COS2RO	$\cos 2\rho$	See symbol	/DYNA	/(120)	AL4 AL7 AL8 NLORV STATEF	I I I I O	COS2RO COS2RO COS2RO COS2RO COS2RO
COS2GM	$\cos 2\gamma$	See symbol	/DYNA	/(121)	AL4 STATEF	I O	COS2GM COS2GM
CM	C_m	Moment coefficient	/DYNA	/(122)	MOMECO UT	O I	CM CM
CMA	C_{m_α}	Moment coefficient slope (RAD^{-1})	/DYNA	/(123)	MOMECO STATEF UT	I M I	CMA CMA CMA
CMM	$\partial C_m/\partial M$	See symbol	/DYNA	/(124)	MOMECO UT	O I	CMM CMM
CMAA	$\partial C_{m_\alpha}/\partial \alpha$	See symbol	/DYNA	/(125)	MOMECO UT	O I	CMAA CMAA
CMMM	$\partial^2 C_m/\partial M^2$	See symbol	/DYNA	/(126)	MOMECO UT	O I	CMMM CMMM
CMAM	$\partial C_{m_\alpha}/\partial M$	See symbol	/DYNA	/(127)	MOMECO STATEF UT	I M I	CMAM CMAM CMAM
CMO	C_{m_0}	Moment coefficient at $\alpha = 0$.	/DYNA	/(128)	MOMECO STATEF	I I	CMO CMO
CMOM	$\partial C_{m_0}/\partial M$	See symbol	/DYNA	/(129)	MOMECO STATEF	I I	CMOM CMOM
CMOMM	$\partial^2 C_{m_0}/\partial M^2$	See symbol	/DYNA	/(130)	MOMECO STATEF	I I	CMOMM CMOMM
CMAMM	$\partial^2 C_{m_\alpha}/\partial M^2$	See symbol	/DYNA	/(131)	MOMECO STATEF	I M	CMAMM CMAMM
ULFTV	$\partial L_\alpha/\partial V$	See symbol	/DYNA	/(132)	AL3 UT	I M	ULFTV ULFTV
ULFTR	$\partial L_\alpha/\partial R$	See symbol	/DYNA	/(133)	AL3 UT	I M	ULFTR ULFTR
ULFTVV	$\partial^2 L_\alpha/\partial V^2$	See symbol	/DYNA	/(134)	AL3 UT	I M	ULFTVV ULFTVV
ULFTVR	$\partial^2 L_\alpha/\partial V \partial R$	See symbol	/DYNA	/(135)	AL3 UT	I M	ULFTVR ULFTVR
ULFTVA	$\partial^2 L_\alpha/\partial V \partial \alpha$	See symbol	/DYNA	/(136)	AL3 UT	I M	ULFTVA ULFTVA
ULFTRR	$\partial^2 L_\alpha/\partial R^2$	See symbol	/DYNA	/(137)	AL3 UT	I M	ULFTRR ULFTRR
ULFTRA	$\partial^2 L_\alpha/\partial R \partial \alpha$	See symbol	/DYNA	/(138)	AL3 UT	I M	ULFTRA ULFTRA
IPOW		Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag	/DYNA	/(139)	ARCIN FORCES NPLANE STATEF THROTL	M I I I I	IPOW IPOW IPOW IPOW IPOW
XARC		Quasitime at which present subarc commenced.	/DYNA	/(140)	ARCIN STATEF	O I	XARC XARC
TSTART		Trajectory time at which present subarc commenced.	/DYNA	/(141)	ARCEN ARCIN STATEF TRAJIN	M M I O	TSTART TSTART TSTART TSTART

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
GH	$\partial g/\partial R$	See symbol	/DYNA	/(142)	AL7	I	GH
					AL8	I	GH
					NLDRV	I	GH
					STATEF	M	GH
GRR	$\partial^2 g/\partial R^2$	See symbol	/DYNA	/(143)	AL7	I	GRR
					AL8	I	GRR
					NLDRV	I	GRR
					STATEF	M	GRR
LIFTAA	$\partial^2 L/\partial \alpha^2$	See symbol	/DYNA	/(144)	AL1	I	LIFTAA
					AL4	I	LIFTAA
					AL5	I	LIFTAA
					AL6	I	LIFTAA
					APPLY	I	LIFTAA
					TH3	I	LIFTAA
					UT	O	LIFTAA
CODMM	$\partial^2 C_{D_0}/\partial M^2$	See symbol	/DYNA	/(145)	AEROCO	I	CODMM
					STATEF	I	CODMM
CLAMM	$\partial^2 C_{L_\alpha}/\partial M^2$	See symbol	/DYNA	/(146)	AEROCO	I	CLAMM
					STATEF	M	CLAMM
CLOM	$\partial C_{L_0}/\partial M$	See symbol	/DYNA	/(147)	AEROCO	I	CLOM
					STATEF	I	CLOM
CLOMM	$\partial^2 C_{L_0}/\partial M^2$	See symbol	/DYNA	/(148)	AEROCO	I	CLOMM
					STATEF	I	CLOMM
DYN149		Not used.	/DYNA	/(149)			
CT	C_T	Value for thrust in case constant thrust constraint is used. (LBS)	/DYNA	/(150)	TH1	I	CT
CODAE	$\cos(\alpha-\delta_E)$	See symbol	/DYNA	/(151)	AL1	I	CODAE
					AL4	I	CODAE
					AL6	I	CODAE
					AL7	I	CODAE
					AL8	I	CODAE
					AL9	I	CODAE
					APPLY	I	CODAE
					CONTRL	I	CODAE
					NLDRV	I	CODAE
					TH3	I	CODAE
					UT	O	CODAE
SIDAE	$\sin(\alpha-\delta_E)$	See symbol	/DYNA	/(152)	AL1	I	SIDAE
					AL4	I	SIDAE
					AL6	I	SIDAE
					AL7	I	SIDAE
					AL8	I	SIDAE
					AL9	I	SIDAE
					APPLY	I	SIDAE
					CONTRL	I	SIDAE
					TH3	I	SIDAE
					UT	O	SIDAE
COD	$\cos\delta_E$	See symbol	/DYNA	/(153)	DL2	I	COD
					OUTPUT	I	COD
					TH3	I	COD
					UT	M	COD
SID	$\sin\delta_E$	See symbol	/DYNA	/(154)	DL2	I	SID
					OUTPUT	I	SID
					TH3	I	SID
					UT	M	SID
DELTAE	δ_E	Engine deflection (RADS)	/DYNA	/(155)	ALGCON	M	DELTAE
					ARCIN	M	DELTAE
					CONTRL	M	DELTAE
					DL1	I	DELTAE
					OUTPUT	I	DELTAE
					TRAJIN	O	DELTAE
					UT	I	DELTAE
CDE	C_{δ_E}	Value for engine deflection in case constant engine deflection constraint is used (RADS)	/DYNA	/(156)	DL1	I	CDE

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XCG	X_{CG}	Center of gravity body x station (FT)	/DYNA	/(157)	DL2	I	XCG
					STATEF	I	XCG
					UT	I	XCG
ZCG	Z_{CG}	Center of gravity body z station (FT)	/DYNA	/(158)	DL2	I	ZCG
					STATEF	I	ZCG
					UT	I	ZCG
XJ	$\dot{j}$	Control blend factor	/DYNA	/(159)	ARCIN	O	XJ
					DL2	I	XJ
					OUTPUT	I	XJ
					STATEF	I	XJ
					UT	I	XJ
XMCG	M_{CG}	Aerodynamic moment about center of gravity (FT-LBS)	/DYNA	/(160)	DL2	I	XMCG
					OUTPUT	I	XMCG
					UT	M	XMCG
CALPHA	C_{α}	Value for angle of attack in case constant angle of attack constraint is used. (RADS)	/DYNA	/(161)	AL2	I	CALPHA
					NPLANE	M	CALPHA
ALMAX	α_{max}	Magnitude of angle of attack constarint (RADS)	/DYNA	/(162)	ARCIN	O	ALMAX
					NPLANE	I	ALMAX
DB	D_b	Base drag (LBS)	/DYNA	/(163)	AL1	I	DB
					AL4	I	DB
					AL6	I	DB
					AL7	I	DB
					AL8	I	DB
					AL9	I	DB
					APPLY	I	DB
					CONTRL	I	DB
					NLDRV	I	DB
					OUTPUT	I	DB
					STATEF	I	DB
					TH3	I	DB
					UT	I	DB
ULFT	L_u	Untrimmed aerodynamic lift (LBS)	/DYNA	/(164)	AL3	I	ULFT
					NPLANE	I	ULFT
					UT	M	ULFT
CULFT	C_{L_u}	Magnitude of untrimmed lift limit (LBS)	/DYNA	/(165)	AL3	I	CULFT
					NPLANE	O	CULFT
ULFTA	$\partial L_u/\partial\alpha$	See symbol	/DYNA	/(166)	AL3	I	ULFTA
					UT	M	ULFTA
TSTAGE		Trajectory time at which present rocket engine ignited. (SECS)	/DYNA	/(167)	ARCIN	M	TSTAGE
					STATEF	I	TSTAGE
					TRAJIN	O	TSTAGE
TIMES		Elapsed burning time of present rocket engine (SECS)	/DYNA	/(168)	STATEF	M	TIMES
XMCGAA	$\partial^2 M_{CG}/\partial\alpha^2$	See symbol	/DYNA	/(169)	DL2	I	XMCGAA
					UT	M	XMCGAA
IRATED		1% of the maximum rated I_{SP} (SECS)	/DYNA	/(170)	ARCIN	O	IRATED
					IMPULS	I	IRATED
FRATED		1% of the maximum rocket vacuum thrust (LBS)	/DYNA	/(171)	ARCIN	O	FRATED
					IMPULS	I	FRATED
MTT		Table number for tabulated rocket vacuum thrust	/DYNA	/(172)	ARCIN	M	MTT
					STATEF	I	MTT
J1		Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA	/(173)	APPLY	I	J1
					ARCIN	O	J1
					CONTRL	M	J1
					FORCES	I	J1
					NPLANE	I	J1
					STATEF	I	J1
					THROTL	M	J1
J2		Engine deflection option flag. J2 = 1: Constant engine deflection; J2 = 2: Moment balancing.	/DYNA	/(174)	ARCIN	O	J2
					CONTRL	I	J2
					NPLANE	I	J2

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
J3		Angle of attack option flag. J3 = 1: Optimal angle of attack; J3 = 2: Constant angle of attack; J3 = 3: Untrimmed lift limit; J3 = 4: Vertical rise or pitchover; J3 = 5: Unpowered total acceleration limit; J3 = 6: Gravity turn; J3 = 7: Dynamic pressure constraint; J3 = 8: Heating rate constraint; J3 = 9: Reynolds number constraint.	/DYNA	/(175)	ARCIN CONTRL NPLANE OUTPUT	O M M I	J3 J3 J3 J3
XMCGA	$\partial M_{CG}/\partial \alpha$	See symbol	/DYNA	/(176)	DL2 UT	I M	XMCGA XMCGA
FVACF		Not used.	/DYNA	/(177)			
ULFTAA	$\partial^2 L_u/\partial \alpha^2$	See symbol	/DYNA	/(178)	AL3 UT	I M	ULFTAA ULFTAA
ISPF	$\partial I_{SP}/\partial T$	See symbol	/DYNA	/(179)	APPLY IMPULS	I O	ISPF ISPF
ISPFF	$\partial^2 I_{SP}/\partial T^2$	See symbol	/DYNA	/(180)	APPLY IMPULS	I O	ISPFF ISPFF
ILOAD		Logical flag that is true if there is any aerodynamic load on the vehicle.	/DYNA	/(181)	ARCIN CONTRL NPLANE UT	M I I I	ILOAD ILOAD ILOAD ILOAD
FKM	$\partial k/\partial M$	See symbol	/DYNA	/(182)	AEROCO STATEF	I I	FKM FKM
FKMM	$\partial^2 k/\partial M^2$	See symbol	/DYNA	/(183)	AEROCO STATEF	I I	FKMM FKMM
SWITCH		Logical flag that is true if this is the compute point at which the powered acceleration constraint commences.	/DYNA	/(184)	CONTRL NPLANE THROTL	I I O	SWITCH SWITCH SWITCH
INQF		State variable inequality constraint flag. INQF = 0: No SVIC in effect; INQF = 7: Dynamic pressure IC in effect; INQF = 8: Heating rate SVIC in effect; INQF = 9: Reynolds number SVIC in effect.	/DYNA	/(185)	ARCIN NPLANE	M M	INQF INQF
CL	C_L	Lift coefficient	/DYNA	/(186)	AEROCO OUTPUT UT	M I I	CL CL CL
CLA	C_{L_α}	Lift coefficient slope (RAD^{-1})	/DYNA	/(187)	AEROCO STATEF UT	M M I	CLA CLA CLA
CLM	$\partial C_L/\partial M$	See symbol	/DYNA	/(188)	AEROCO UT	M I	CLM CLM
CLAA	$\partial C_{L_\alpha}/\partial \alpha$	See symbol	/DYNA	/(189)	AEROCO UT	M I	CLAA CLAA
CLMM	$\partial^2 C_L/\partial M^2$	See symbol	/DYNA	/(190)	AEROCO UT	M I	CLMM CLMM
CLAM	$\partial C_{L_\alpha}/\partial M$	See symbol	/DYNA	/(191)	AEROCO STATEF UT	M M I	CLAM CLAM CLAM
CD	C_D	Drag coefficient	/DYNA	/(192)	AEROCO OUTPUT UT	O I I	CD CD CD
CDA	C_{D_α}	Drag coefficient slope (RAD^{-1})	/DYNA	/(193)	AEROCO UT	M I	CDA CDA
CDM	$\partial C_D/\partial M$	See symbol	/DYNA	/(194)	AEROCO UT	O I	CDM CDM
CDAA	$\partial C_{D_\alpha}/\partial \alpha$	See symbol	/DYNA	/(195)	AEROCO UT	M I	CDAA CDAA
CDMM	$\partial^2 C_D/\partial M^2$	See symbol	/DYNA	/(196)	AEROCO UT	O I	CDMM CDMM

Ecific fuel consumption of airbreather

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CDAM	$\partial C_{D_m}/\partial M$	See symbol	/DYNA /(	197)	AEROCO UT	M I	CDAM CDAM
DYN198		Not used.	/DYNA /(	198)			
DYN199		Not used.	/DYNA /(	199)			
DYN200		Not used.	/DYNA /(	200)			
XMCGV	$\partial M_{CG}/\partial V$	See symbol	/DYNA /(	201)	DL2 UT	I M	XMCGV XMCGV
XMCGR	$\partial M_{CG}/\partial R$	See symbol	/DYNA /(	202)	DL2 UT	I M	XMCGR XMCGR
XMCGM	$\partial M_{CG}/\partial m$	See symbol	/DYNA /(	203)	DL2 UT	I M	XMCGM XMCGM
XMCGVV	$\partial^2 M_{CG}/\partial V^2$	See symbol	/DYNA /(	204)	DL2 UT	I M	XMCGVV XMCGVV
XMCGVR	$\partial^2 M_{CG}/\partial V \partial R$	See symbol	/DYNA /(	205)	DL2 UT	I M	XMCGVR XMCGVR
XMCGVM	$\partial^2 M_{CG}/\partial V \partial m$	See symbol	/DYNA /(	206)	DL2 UT	I M	XMCGVM XMCGVM
XMCGVA	$\partial^2 M_{CG}/\partial V \partial \alpha$	See symbol	/DYNA /(	207)	DL2 UT	I M	XMCGVA XMCGVA
XMCGRR	$\partial^2 M_{CG}/\partial \alpha^2$	See symbol	/DYNA /(	208)	DL2 UT	I M	XMCGRR XMCGRR
XMCGRM	$\partial^2 M_{CG}/\partial R \partial m$	See symbol	/DYNA /(	209)	DL2 UT	I M	XMCGRM XMCGRM
XMCGRA	$\partial^2 M_{CG}/\partial R \partial \alpha$	See symbol	/DYNA /(	210)	DL2 UT	I M	XMCGRA XMCGRA
XMCGMM	$\partial^2 M_{CG}/\partial m^2$	See symbol	/DYNA /(	211)	DL2 UT	I M	XMCGMM XMCGMM
XMCGMA	$\partial^2 M_{CG}/\partial m \partial \alpha$	See symbol	/DYNA /(	212)	DL2 UT	I M	XMCGMA XMCGMA
RORRR	$\partial^3 \rho_a/\partial R^3$	See symbol	/DYNA /(	213)	AL7 AL8 AL9 STATEF	I I I I	RORRR RORRR RORRR RORRR
DYN214	μ_a	Atmospheric viscosity (dynamic) (SLGS/FT/SEC)	/DYNA /(	214)	OUTPUT PDBCOL	I I	VMU VMU
DYN215	$\partial \mu_a/\partial R$	See symbol	/DYNA /(	215)	PDBCOL	I	VMR
DYN216	$\partial^2 \mu_a/\partial R^2$	See symbol	/DYNA /(	216)			
DYN217	$\partial^3 \mu_a/\partial R^3$	See symbol	/DYNA /(	217)			
IDAM		Optional atmospheric calculations flag. IDAM = -1: Compute $\partial^3 \rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3 \rho_a/\partial R^3$, μ_a, $\partial \mu_a/\partial R$, etc.	/DYNA /(	218)	ARCIN ERROR NPLANE STATEF WRAPUP	O I O M O	IDAM IDAM IDAM IDAM IDAM
TAIRB		Air-breather engine thrust (LBS)	/DYNA /(	219)	STATEF TH4	I I	TAIRB TAIRB
TAIRBV		Partial of TAIRB wrt V	/DYNA /(	220)	TH4	I	TAIRBV
TAIRBH		Partial of TAIRB wrt h (altitude)	/DYNA /(	221)	TH4	I	TAIRBH
TARBVV		Second partial of TAIRB wrt V	/DYNA /(	222)	TH4	I	TARBVV
TARBHH		Second partial of TAIRB wrt h	/DYNA /(	223)	TH4	I	TARBHH
TARBVH		Second partial of TAIRB wrt V and h	/DYNA /(	224)	TH4	I	TARBVH
SFC			/DYNA /(	225)	APPLY	I	SFC
SFCV		Partial of SFC wrt V	/DYNA /(	226)	APPLY	I	SFCV
SFCH		Partial of SFC wrt h	/DYNA /(	227)	APPLY	I	SFCH
SFCVV		Second partial of SFC wrt V	/DYNA /(	228)	APPLY	I	SFCVV

808

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SFCHH		Second partial of SFC wrt h	/DYNA	/(229)	APPLY	I	SFCHH
SFCVH		Second partial of SFC wrt V and h	/DYNA	/(230)	APPLY	I	SFCVH

BLØCK
EVAL

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SGN		Sign of the variable SIG in the 65th word of common block /GOBAL/. SGN = +: payoff to be maximized; SGN = -: payoff to be minimized.	/EVAL	/(1)	BNDRY ENDPT	D I	SGN SGN
SPART		An 18 word array whose first nine entries receive the values of the partial derivatives wrt the state of those target conditions computed in subroutine PDBCQL.	/EVAL	/(2)	BNDRY BRANPT ENDPT INTRPT	D I I I	SPART SPART SPART SPART
MAP		A 10 word array that maps the steepest descent state vector into the QL state vector.	/EVAL	/(20)	BNDRY BRANPT ENDPT INTRPT	D I I I	MAP MAP MAP MAP
PZI		A 40 word array that contains the target condition misses for all the target conditions in the problem.	/EVAL	/(30)	BNDRY BRANPT ENDPT INTRPT	I M M M	PZI PZI PZI PZI
NOCK	m_I	The number of c's in the vector C_I defined by Equation 17.4-4 of Vol.I of this document.	/EVAL	/(70)	BNDRY BRANPT ENDPT INTRPT	M I I I	NOCK NOCK NOCK NOCK
S		An 18x41 array used to store the particular and homogeneous solutions on the early side of a corner point.	/EVAL	/(71)	BNDRY BRANPT ENDPT INTRPT	I I I I	S S S S
TEMP	$(\partial\psi_I/\partial C_I)^T$	A 40 word array that contains the transpose of the vector defined by Equation 17.4-9 of Vol.I of this document.	/EVAL	/(809)	BRANPT ENDPT INTRPT	I I I	TEMP TEMP TEMP
DZ	$\Delta c_i h_i(I^-)$	An 18 word array that contains the second term on the right hand side of Equation 17.4-11 of Vol.I of this document.	/EVAL	/(849)	BRANPT ENDPT INTRPT	I I I	DZ DZ DZ
DC	Δc_i	Small perturbation of a c.	/EVAL	/(867)	BNDRY BRANPT ENDPT INTRPT	D I I I	DC DC DC DC
L		Total number of target conditions to satisfy in the problem.	/EVAL	/(868)	BNDRY BRANPT ENDPT INTRPT	M M M M	L L L L
SI		An 18x41 array used to store the particular and homogeneous solutions on the late side of a corner point.	/EVAL	/(869)	BRANPT INTRPT	I I	SI SI

BLØCK
F

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
F		An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-7 thru -10 of Vol.I of this document.	/F	/(1)	MADAMS SALVE	I I	F F

598

BLØCK GLØBAL

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
GR	g_r	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5 APPLY BRANPT COSTAB COSTAI INTRPT OUTPUT PDBCQL QLTOSZ SALVE STATEF TM3	I I I I I I I I I I I I	GR GR GR GR GR GR GR GR GR GR GR GR
ER	E_R	Earth radius. (FT)	/GLOBAL/(	2)	ENVPRQ PDBCQL QLTOSZ STATEF	I I I I	ER ER ER ER
OMGZ	ω	Earth rotation rate (RAD/SEC)	/GLOBAL/(	3)	PDBCQL TRAJIN	I I	OMGZ OMGZ
XLAMRF	ρ_r	Reference latitude. (DEG)	/GLOBAL/(	4)	CHECK	I	XLAMRF
YMURF	μ_r	Reference longitude. (DEG)	/GLOBAL/(	5)	CHECK	I	YMURF
LUM		Program control flag. LUM = 0: Steepest descent only; LUM = 1: Steepest descent and adjoint transformation stored on tape; LUM = 2: Steepest descent and QL; LUM = 3: QL only.	/GLOBAL/(	6)			
TO	t_0	Trajectory start time. (SEC)	/GLOBAL/(	7)	FETCH INARC TRAJIN WRAPUP	M M I I	TO TO TO TO
EPSLON	ϵ	QL iteration convergence criterion.	/GLOBAL/(	8)	CHECK GROPE	M I	EPSLON EPSLON
INNER		Number of Adams-Moulton inner loops.	/GLOBAL/(	9)	CHECK MADAMS	M I	INNER INNER
ITRMAX		Maximum number of QL iterations.	/GLOBAL/(	10)	CHECK GROPE	M I	ITRMAX ITRMAX
JJOP		A six word array used for various internal flags.	/GLOBAL/(	11)			
IFATAL		Fatal error flag.	/GLOBAL/(	17)	ERROR	O	IFATAL
NARC	N_3	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND BNDRY CHECK ENDPT ENVPRQ FETCH INARC MAGIC QLTOSZ SALVE WRAPUP	I I I I I I I I I I I	NARC NARC NARC NARC NARC NARC NARC NARC NARC NARC NARC
NBRAN	N_1	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY BRANPT COSTAB ENVPRQ INTRPT MAGIC QLTOSZ SALVE	I I I I I I I I	NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
NFARC	N_2	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND BNDRY BRANPT COSTAB ENVPRO INTRPT MAGIC QLTOSZ SALVE	I I I I I I I I I	NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC
ID		A four word array containing the basic deck, reference run, case and part case numbers in that order.	/GLOBAL/(	21)			
KTAB		A 20 word array containing the number of state target conditions specified at the end of each subarc.	/GLOBAL/(	25)	BCOND	I	KTAB
ITAB		A 20 word array containing the number of nonzero state initial conditions specified at the beginning of each subarc.	/GLOBAL/(	45)	BCOND	I	ITAB
SIG		Payoff sign. SIG < 0: Payoff to be minimized; SIG > 0: Payoff to be maximized.	/GLOBAL/(	65)	BNDRY	I	SIG
MAXTAB		Largest univariant table number in this case.	/GLOBAL/(	66)	SPLINE	I	NT
GM	GM	Product of Newton's universal gravitational constant and the mass of the earth. (FT^3/SEC^2)	/GLOBAL/(	67)	OUTPUT PDBCQL STATEF	I I I	GM GM GM
PSIRF	ψ_r	Reference azimuth. (DEG)	/GLOBAL/(	68)	CHECK	I	PSIRF
IPFLG1		IPFLG1≠0 supresses print-out of velocity losses and inertial Euler angles.	/GLOBAL/(	69)	OUTPUT PDBCQL QLTOSZ SALVE	I I O I	IPFLG1 IPFLG1 IPFLG1 IPFLG1
IPFLG2		IPFLG2≠0 supresses print-out of orbital parameters.	/GLOBAL/(	70)	QLTOSZ	O	IPFLG2
IPFLG3		IPFLG3≠0 supresses print-out of impact data.	/GLOBAL/(	71)	OUTPUT QLTOSZ	I O	IPFLG3 IPFLG3
IPFLG4		IPFLG4≠0 supresses print-out of inertial Cartesian coordinates.	/GLOBAL/(	72)			
INEQFL		A 20 word array that contains the code number of the state variables inequality constraint that applies on each subarc. A zero entry indicates that no SVIC applies.	/GLOBAL/(	73)	ARCIN	I	INEQFL
ITPSO		A non zero input value indicates to the steepest descent module that the initial steering angle profiles are stored on logical unit 11.	/GLOBAL/(	93)			
KSOL		An internal flag that has the same significance as ITPSO.	/GLOBAL/(	94)			
INARK		Logical unit on which initial and converged arcs are stored. INARK = 11.	/GLOBAL/(	95)	CHECK FETCH INARC MARCH WRAPUP	O I I I I	INARK INARK INARK INARK INARK
KGLOBL		A seven word array not used.	/GLOBAL/(	96)	ERROR	O	KGLOBL

BLØCK
JACØB

808

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
JAKE		An 18x18 array defined by Equation 17.5-5 in Vol.I of this document. The entry in the i-th row and j-th column is the total partial derivative of the quasitime derivative of the Y_i component of Y with respect to the Y_j component of Y, i.e., $\partial \dot{Y}_i / \partial Y_j$, where $Y^T = (y^T, \lambda^T)$	/JACOB /(	1)	LINDRV NLDRV NLDRV SALVE	I M M O	JAKE JACOB VDV JAKE
GDV		Description not input	/JACOB /(	2)	NLDRV	M	GDV
PDV		Description not input	/JACOB /(	3)	NLDRV	M	PDV
RDV		Description not input	/JACOB /(	4)	NLDRV	M	RDV
ODV		Description not input	/JACOB /(	5)	NLDRV	M	ODV
UDV		Description not input	/JACOB /(	6)	NLDRV	M	UDV
MDV		Description not input	/JACOB /(	7)	NLDRV	M	MDV
HTDV		Description not input	/JACOB /(	9)	NLDRV	M	HTDV
LVDV		Description not input	/JACOB /(	10)	NLDRV	M	LVDV
LGDV		Description not input	/JACOB /(	11)	NLDRV	O	LGDV
LPDV		Description not input	/JACOB /(	12)	NLDRV	O	LPDV
LRDV		Description not input	/JACOB /(	13)	NLDRV	O	LRDV
LODV		Description not input	/JACOB /(	14)	NLDRV	O	LODV
LMDV		Description not input	/JACOB /(	16)	NLDRV	O	LMDV
LTDV		Description not input	/JACOB /(	17)	NLDRV	O	LTDV
VDG		Description not input	/JACOB /(	19)	NLDRV	M	VDG
GDG		Description not input	/JACOB /(	20)	NLDRV	M	GDG
PDG		Description not input	/JACOB /(	21)	NLDRV	M	PDG
RDG		Description not input	/JACOB /(	22)	NLDRV	M	RDG
ODG		Description not input	/JACOB /(	23)	NLDRV	M	ODG
UDG		Description not input	/JACOB /(	24)	NLDRV	M	UDG
MDG		Description not input	/JACOB /(	25)	NLDRV	M	MDG
LVDG		Description not input	/JACOB /(	28)	NLDRV	M	LVDG
LGDG		Description not input	/JACOB /(	29)	NLDRV	M	LGDG
LPDG		Description not input	/JACOB /(	30)	NLDRV	O	LPDG
LRDG		Description not input	/JACOB /(	31)	NLDRV	O	LRDG
LODG		Description not input	/JACOB /(	32)	NLDRV	O	LODG
LMDG		Description not input	/JACOB /(	34)	NLDRV	O	LMDG
LTDG		Description not input	/JACOB /(	35)	NLDRV	O	LTDG
VDP		Description not input	/JACOB /(	37)	NLDRV	M	VDP
GDP		Description not input	/JACOB /(	38)	NLDRV	M	GDP
PDP		Description not input	/JACOB /(	39)	NLDRV	M	PDP
ODP		Description not input	/JACOB /(	41)	NLDRV	M	ODP
UDP		Description not input	/JACOB /(	42)	NLDRV	M	UDP
MDP		Description not input	/JACOB /(	43)	NLDRV	M	MDP
LVDP		Description not input	/JACOB /(	46)	NLDRV	M	LVDP
LGDP		Description not input	/JACOB /(	47)	NLDRV	M	LGDP
LPDP		Description not input	/JACOB /(	48)	NLDRV	M	LPDP
LRDP		Description not input	/JACOB /(	49)	NLDRV	O	LRDP
LODP		Description not input	/JACOB /(	50)	NLDRV	O	LODP
LTDP		Description not input	/JACOB /(	53)	NLDRV	O	LTDP
VDR		Description not input	/JACOB /(	55)	NLDRV	M	VDR
GDR		Description not input	/JACOB /(	56)	NLDRV	M	GDR

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
PDR		Description not input	/JACOB /(	57)	NLDRV	M	PDR
ODR		Description not input	/JACOB /(	59)	NLDRV	M	ODR
UDR		Description not input	/JACOB /(	60)	NLDRV	M	UDR
MDR		Description not input	/JACOB /(	61)	NLDRV	M	MDR
HTDR		Description not input	/JACOB /(	63)	NLDRV	M	HTDR
LVDR		Description not input	/JACOB /(	64)	NLDRV	M	LVDR
LGDR		Description not input	/JACOB /(	65)	NLDRV	M	LGDR
LPDR		Description not input	/JACOB /(	66)	NLDRV	M	LPDR
LRDR		Description not input	/JACOB /(	67)	NLDRV	M	LRDR
LODR		Description not input	/JACOB /(	68)	NLDRV	O	LODR
LMDR		Description not input	/JACOB /(	70)	NLDRV	O	LMDR
LTDR		Description not input	/JACOB /(	71)	NLDRV	O	LTDR
VDO		Description not input	/JACOB /(	73)	NLDRV	M	VDO
GDO		Description not input	/JACOB /(	74)	NLDRV	M	GDO
PDO		Description not input	/JACOB /(	75)	NLDRV	M	PDO
UDO		Description not input	/JACOB /(	78)	NLDRV	M	UDO
MDO		Description not input	/JACOB /(	79)	NLDRV	M	MDO
LVDO		Description not input	/JACOB /(	82)	NLDRV	M	LVDO
LGDO		Description not input	/JACOB /(	83)	NLDRV	M	LGDO
LPDO		Description not input	/JACOB /(	84)	NLDRV	M	LPDO
LRDO		Description not input	/JACOB /(	85)	NLDRV	M	LRDO
LODO		Description not input	/JACOB /(	86)	NLDRV	M	LODO
LTDO		Description not input	/JACOB /(	89)	NLDRV	O	LTDO
VDM		Description not input	/JACOB /(	109)	NLDRV	M	VDM
GDM		Description not input	/JACOB /(	110)	NLDRV	M	GDM
PDM		Description not input	/JACOB /(	111)	NLDRV	M	PDM
MDM		Description not input	/JACOB /(	115)	NLDRV	M	MDM
LVDM		Description not input	/JACOB /(	118)	NLDRV	O	LVDM
LGDM		Description not input	/JACOB /(	119)	NLDRV	O	LGDM
LPDM		Description not input	/JACOB /(	120)	NLDRV	O	LPDM
LRDM		Description not input	/JACOB /(	121)	NLDRV	O	LRDM
LODM		Description not input	/JACOB /(	122)	NLDRV	O	LODM
LMDM		Description not input	/JACOB /(	124)	NLDRV	O	LMDM
LTDM		Description not input	/JACOB /(	125)	NLDRV	O	LTDM
VDT		Description not input	/JACOB /(	127)	NLDRV	M	VDT
GDT		Description not input	/JACOB /(	128)	NLDRV	M	GDT
PDT		Description not input	/JACOB /(	129)	NLDRV	M	PDT
RDT		Description not input	/JACOB /(	130)	NLDRV	M	RDT
ODT		Description not input	/JACOB /(	131)	NLDRV	M	ODT
UDT		Description not input	/JACOB /(	132)	NLDRV	M	UDT
MDT		Description not input	/JACOB /(	133)	NLDRV	M	MDT
HTDT		Description not input	/JACOB /(	135)	NLDRV	M	HTDT
LVDT		Description not input	/JACOB /(	136)	NLDRV	M	LVDT
LGDT		Description not input	/JACOB /(	137)	NLDRV	M	LGDT
LPDT		Description not input	/JACOB /(	138)	NLDRV	M	LPDT
LRDT		Description not input	/JACOB /(	139)	NLDRV	M	LRDT
LODT		Description not input	/JACOB /(	140)	NLDRV	M	LODT
LMDT		Description not input	/JACOB /(	142)	NLDRV	O	LMDT
LTDT		Description not input	/JACOB /(	143)	NLDRV	O	LTDT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
VDLV		Description not input	/JACOB /(	163)	NLDRV	0	VDLV
GDLV		Description not input	/JACOB /(	164)	NLDRV	0	GDLV
PDLV		Description not input	/JACOB /(	165)	NLDRV	0	PDLV
MDLV		Description not input	/JACOB /(	169)	NLDRV	0	MDLV
LVDLV		Description not input	/JACOB /(	172)	NLDRV	0	LVDLV
LGDLV		Description not input	/JACOB /(	173)	NLDRV	0	LGDLV
LPDLV		Description not input	/JACOB /(	174)	NLDRV	0	LPDLV
LRDLV		Description not input	/JACOB /(	175)	NLDRV	0	LRDLV
LODLV		Description not input	/JACOB /(	176)	NLDRV	0	LODLV
LMDLV		Description not input	/JACOB /(	178)	NLDRV	0	LMDLV
LTDLV		Description not input	/JACOB /(	179)	NLDRV	0	LTDLV
VDLG		Description not input	/JACOB /(	181)	NLDRV	0	VDLG
GDLG		Description not input	/JACOB /(	182)	NLDRV	0	GDLG
PDLG		Description not input	/JACOB /(	183)	NLDRV	0	PDLG
MDLG		Description not input	/JACOB /(	187)	NLDRV	0	MDLG
LVDLG		Description not input	/JACOB /(	190)	NLDRV	0	LVDLG
LGDLG		Description not input	/JACOB /(	191)	NLDRV	0	LGDLG
LPDLG		Description not input	/JACOB /(	192)	NLDRV	0	LPDLG
LRDLG		Description not input	/JACOB /(	193)	NLDRV	0	LRDLG
LODLG		Description not input	/JACOB /(	194)	NLDRV	0	LODLG
LMDLG		Description not input	/JACOB /(	196)	NLDRV	0	LMDLG
LTDLG		Description not input	/JACOB /(	197)	NLDRV	0	LTDLG
VDLP		Description not input	/JACOB /(	199)	NLDRV	0	VDLP
GDLP		Description not input	/JACOB /(	200)	NLDRV	0	GDLP
PDLP		Description not input	/JACOB /(	201)	NLDRV	0	PDLP
MDLP		Description not input	/JACOB /(	205)	NLDRV	0	MDLP
LVDLP		Description not input	/JACOB /(	208)	NLDRV	0	LVDLP
LGDLP		Description not input	/JACOB /(	209)	NLDRV	0	LGDLP
LPDLP		Description not input	/JACOB /(	210)	NLDRV	0	LPDLP
LRDLP		Description not input	/JACOB /(	211)	NLDRV	0	LRDLP
LODLP		Description not input	/JACOB /(	212)	NLDRV	0	LODLP
LMDLP		Description not input	/JACOB /(	214)	NLDRV	0	LMDLP
LTDLP		Description not input	/JACOB /(	215)	NLDRV	0	LTDLP
LVDLR		Description not input	/JACOB /(	226)	NLDRV	0	LVDLR
LGDLR		Description not input	/JACOB /(	227)	NLDRV	0	LGDLR
LTDLR		Description not input	/JACOB /(	233)	NLDRV	0	LTDLR
LVDLO		Description not input	/JACOB /(	244)	NLDRV	0	LVDLO
LGDLO		Description not input	/JACOB /(	245)	NLDRV	0	LGDLO
LPDLO		Description not input	/JACOB /(	246)	NLDRV	0	LPDLO
LRDLO		Description not input	/JACOB /(	247)	NLDRV	0	LRDLO
LTDLO		Description not input	/JACOB /(	251)	NLDRV	0	LTDLO
LVDLU		Description not input	/JACOB /(	262)	NLDRV	0	LVDLU
LGDLU		Description not input	/JACOB /(	263)	NLDRV	0	LGDLU
LPDLU		Description not input	/JACOB /(	264)	NLDRV	0	LPDLU
LRDLU		Description not input	/JACOB /(	265)	NLDRV	0	LRDLU
LODLU		Description not input	/JACOB /(	266)	NLDRV	0	LODLU
LTDLU		Description not input	/JACOB /(	269)	NLDRV	0	LTDLU
LVDLM		Description not input	/JACOB /(	280)	NLDRV	0	LVDLM
LRDLM		Description not input	/JACOB /(	283)	NLDRV	0	LRDLM

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LMDLM		Description not input	/JACOB /(	286)	NLDRV	0	LMDLM
LTDLM		Description not input	/JACOB /(	287)	NLDRV	0	LTDLM
LVDLH		Description not input	/JACOB /(	316)	NLDRV	0	LVDLH
LRDLH		Description not input	/JACOB /(	319)	NLDRV	0	LRDLH
LTDLH		Description not input	/JACOB /(	323)	NLDRV	0	LTDLH

10 NOV 72 6.01-47

BLØCK
LASTAB

814

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LASTAB		Description not input	/LASTAB/(	1)	TIMFD	I	LASTAB

BLØCK
MAP

9/8

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
MAP		An array that maps the initial arc state and costate into the QL state and costate.	/MAP	/(1)	CHECK INARC	O I	MAP MAP

BLØCK
MATS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
P1		First entry of 3 word in-plane control vector m	/MATS	/(1)	ALGCON	M	P
					ALGCON	M	P1
P2		Description not input	/MATS	/(2)	ALGCON	M	P2
P3		Description not input	/MATS	/(3)	ALGCON	M	P3
XK1		First entry of 3 word in-plane control constraints K	/MATS	/(4)	ALGCON	I	XK1
					THROTL	I	XK1
					TH1	O	XK1
					TH2	O	XK1
					TH3	O	XK1
					TH4	O	XK1
XK2		Description not input	/MATS	/(5)	DL1	O	XK2
					DL2	O	XK2
XK3		Description not input	/MATS	/(6)	AL1	O	XK3
					AL2	O	XK3
					AL3	O	XK3
					AL4	O	XK3
					AL5	O	XK3
					AL6	O	XK3
					AL7	O	XK3
					AL8	O	XK3
					NPLANE	I	XK3
					OUTPUT	I	XK3
XK1T	$K_T^{(1)}$	First entry of 3x3 matrix containing the explicit partials of K with respect to m, K_m	/MATS	/(7)	ALGCON	I	XK1T
					TH1	O	XK1T
					TH2	O	XK1T
					TH3	M	XK1T
XK2T		Description not input.	/MATS	/(8)	ALGCON	I	XK2T
					DL2	O	XK2T
XK3T		Description not input	/MATS	/(9)	ALGCON	I	XK3T
					AL1	O	XK3T
					AL4	O	XK3T
					AL6	O	XK3T
					AL7	O	XK3T
					AL8	O	XK3T
					AL9	O	XK3T
XK1D		Description not input	/MATS	/(10)	ALGCON	I	XK1D
					TH3	M	XK1D
XK2D		Description not input	/MATS	/(11)	ALGCON	I	XK2D
					AL1	I	XK2D
					DL1	O	XK2D
					DL2	O	XK2D
XK3D		Description not input	/MATS	/(12)	ALGCON	I	XK3D
					AL1	O	XK3D
					AL4	O	XK3D
					AL6	O	XK3D
					AL7	O	XK3D
					AL8	O	XK3D
					AL9	O	XK3D
XK1A		Description not input	/MATS	/(13)	ALGCON	I	XK1A
					TH3	M	XK1A
XK2A		Description not input	/MATS	/(14)	ALGCON	I	XK2A
					AL1	I	XK2A
					DL2	O	XK2A
XK3A		Description not input	/MATS	/(15)	ALGCON	I	XK3A
					AL1	O	XK3A
					AL2	O	XK3A
					AL3	O	XK3A
					AL4	O	XK3A
					AL5	M	XK3A
					AL6	O	XK3A
					AL7	O	XK3A
					AL8	O	XK3A
					AL9	O	XK3A
VDA	$K_{\lambda_v}^{(3)}$	Explicit partial of $K^{(3)}$ wrt λ_v when m is optimal.	/MATS	/(16)	ALGCON	I	VDA
					AL1	O	VDA

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
GDA	$K^{(3)}_{\lambda_\gamma}$	Explicit partial of $K^{(3)}$ wrt λ_γ when ω is optimal.	/MATS /(	17)	ALGCON	I	GDA
					AL1	O	GDA
PDA	$K^{(3)}_{\lambda_\psi}$	Explicit partial of $K^{(3)}$ wrt λ_ψ when ω is optimal.	/MATS /(	18)	ALGCON	I	PDA
					AL1	O	PDA
XM19		Not used	/MATS /(	19)			
XM20		Not used	/MATS /(	20)			
XM21		Not used	/MATS /(	21)			
XM22		Not used	/MATS /(	22)			
XKITT	$K^{(1)}_{TT}$	First entry in 3x6 matrix containing $K_{\omega\omega}$	/MATS /(	23)	ALGCON	I	XKWW
					TH3	O	XK1TT
XK1TD		Description not input	/MATS /(	26)	TH3	M	XK1TD
XK2TD		Description not input	/MATS /(	27)	AL1	I	XK2TD
					DL2	O	XK2TD
XK3TD		Description not input	/MATS /(	28)	AL4	M	XK3TD
					AL6	O	XK3TD
					AL7	M	XK3TD
					AL8	O	XK3TD
					AL9	M	XK3TD
XK1TA		Description not input	/MATS /(	29)	TH3	M	XK1TA
XK2TA		Description not input	/MATS /(	30)	AL1	I	XK2TA
XK3TA		Description not input	/MATS /(	31)	AL4	O	XK3TA
					AL6	O	XK3TA
					AL7	O	XK3TA
					AL8	O	XK3TA
					AL9	O	XK3TA
XK1DD		Description not input	/MATS /(	32)	TH3	M	XK1DD
XK2DD		Description not input	/MATS /(	33)	AL1	I	XK2DD
					DL2	O	XK2DD
XK3DD		Description not input	/MATS /(	34)	AL4	M	XK3DD
					AL6	O	XK3DD
					AL7	M	XK3DD
					AL8	O	XK3DD
					AL9	O	XK3DD
XK1DA		Description not input	/MATS /(	35)	TH3	M	XK1DA
XK2DA		Description not input	/MATS /(	36)	AL1	I	XK2DA
XK3DA		Description not input	/MATS /(	37)	AL4	O	XK3DA
					AL6	O	XK3DA
					AL7	O	XK3DA
					AL8	O	XK3DA
					AL9	M	XK3DA
XK1AA		Description not input	/MATS /(	38)	TH3	M	XK1AA
XK2AA		Description not input	/MATS /(	39)	AL1	I	XK2AA
					DL2	O	XK2AA
XK3AA		Description not input	/MATS /(	40)	AL3	O	XK3AA
					AL4	O	XK3AA
					AL5	M	XK3AA
					AL6	O	XK3AA
					AL7	O	XK3AA
					AL8	O	XK3AA
					AL9	O	XK3AA
XM41		Not used	/MATS /(	41)			
XM42		Not used	/MATS /(	42)			
XM43		Not used	/MATS /(	43)			
XM44		Not used	/MATS /(	44)			
XM45		Not used	/MATS /(	45)			
XK1V	$K^{(1)}_V$	The first entry in a 3x8 matrix containing K_y	/MATS /(	46)	ALGCON	I	XKY
					ALGCON	I	XK1V
					TH3	M	XK1V
					TH4	O	XK1V

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XK2V		Description not input	/MATS /(	47)	DL2	O	XK2V
XK3V		Description not input	/MATS /(	48)	AL1	O	XK3V
					AL3	O	XK3V
					AL4	O	XK3V
					AL5	M	XK3V
					AL6	O	XK3V
					AL7	O	XK3V
					AL8	O	XK3V
					AL9	O	XK3V
XK3G		Description not input	/MATS /(	51)	AL1	O	XK3G
					AL4	O	XK3G
					AL7	O	XK3G
					AL8	O	XK3G
					AL9	O	XK3G
XK3P		Description not input	/MATS /(	54)	AL4	O	XK3P
					AL7	O	XK3P
					AL8	O	XK3P
					AL9	O	XK3P
XK1R		Description not input	/MATS /(	55)	TH2	O	XK1R
					TH3	M	XK1R
					TH4	O	XK1R
XK2R		Description not input	/MATS /(	56)	DL2	O	XK2R
XK3R		Description not input	/MATS /(	57)	AL1	O	XK3R
					AL3	O	XK3R
					AL4	O	XK3R
					AL5	M	XK3R
					AL6	O	XK3R
					AL7	O	XK3R
					AL8	O	XK3R
					AL9	O	XK3R
XK3D		Description not input	/MATS /(	60)	AL4	O	XK3D
					AL7	O	XK3D
					AL8	O	XK3D
					AL9	O	XK3D
XK1M		Description not input	/MATS /(	64)	TH3	M	XK1M
XK2M		Description not input	/MATS /(	65)	DL2	O	XK2M
XK3M		Description not input	/MATS /(	66)	AL1	O	XK3M
					AL4	O	XK3M
					AL5	M	XK3M
					AL6	O	XK3M
					AL7	O	XK3M
					AL8	O	XK3M
					AL9	O	XK3M
XK1Z		Description not input	/MATS /(	67)	TH2	O	XK1Z
XK1VT	$K_{VT}^{(1)}$	The first entry in a 3x3 matrix containing K_{vm}	/MATS /(	70)	ALGCON	I	XKWY
					TH3	M	XK1VT
XK3VT		Description not input	/MATS /(	72)	AL4	O	XK3VT
					AL9	O	XK3VT
XK1VD		Description not input	/MATS /(	73)	TH3	M	XK1VD
XK2VD		Description not input	/MATS /(	74)	AL1	I	XK2VD
XK3VD		Description not input	/MATS /(	75)	AL4	O	XK3VD
					AL9	O	XK3VD
XK1VA		Description not input	/MATS /(	76)	TH3	M	XK1VA
XK2VA		Description not input	/MATS /(	77)	AL1	I	XK2VA
					DL2	O	XK2VA
XK3VA		Description not input	/MATS /(	78)	AL3	O	XK3VA
					AL4	O	XK3VA
					AL5	M	XK3VA
					AL6	O	XK3VA
					AL7	O	XK3VA
					AL8	O	XK3VA
					AL9	O	XK3VA
XK1GT	$K^{(1)}$	The first entry in a 3x3 matrix containing $K_{\gamma m}$	/MATS /(	79)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XK3GT		Description not input	/MATS	/(81)	AL4	O	XK3GT
XK3GD		Description not input	/MATS	/(84)	AL4	O	XK3GD
XK3GA		Description not input	/MATS	/(87)	AL4	O	XK3GA
XK1PT	$K^{(1)}_{\psi T}$	The first entry in a 3x3 matrix containing $K_{\psi m}$	/MATS	/(88)			
XK3PT		Description not input	/MATS	/(90)	AL4	O	XK3PT
XK3PD		Description not input	/MATS	/(93)	AL4	O	XK3PD
XK3PA		Description not input	/MATS	/(96)	AL4	O	XK3PA
XK1RT	$K^{(1)}_{hT}$	The first entry in a 3x3 matrix containing K_{hm}	/MATS	/(97)	TH3	M	XK1RT
XK3RT		Description not input	/MATS	/(99)	AL4	O	XK3RT
					AL7	O	XK3RT
					AL8	O	XK3RT
					AL9	O	XK3RT
XK1RD		Description not input	/MATS	/(100)	TH3	M	XK1RD
XK2RD		Description not input	/MATS	/(101)	AL1	I	XK2RD
XK3RD		Description not input	/MATS	/(102)	AL4	O	XK3RD
					AL7	O	XK3RD
					AL8	O	XK3RD
					AL9	O	XK3RD
XK1RA		Description not input	/MATS	/(103)	TH3	M	XK1RA
XK2RA		Description not input	/MATS	/(104)	AL1	I	XK2RA
					DL2	O	XK2RA
XK3RA		Description not input	/MATS	/(105)	AL3	O	XK3RA
					AL4	O	XK3RA
					AL5	M	XK3RA
					AL6	O	XK3RA
					AL7	O	XK3RA
					AL8	O	XK3RA
					AL9	O	XK3RA
XK1OT	$K^{(1)}_{\rho T}$	The first entry in a 3x3 matrix containing $K_{\rho m}$	/MATS	/(106)			
XK3OT		Description not input	/MATS	/(108)	AL4	O	XK3OT
XK3OD		Description not input	/MATS	/(111)	AL4	O	XK3OD
XK3OA		Description not input	/MATS	/(114)	AL4	O	XK3OA
XK1UT	$K^{(1)}_{\mu T}$	The first entry in a 3x3 matrix containing $K_{\mu m}$	/MATS	/(115)			
XK1MT	$K^{(1)}_{mT}$	The first entry in a 3x3 matrix containing K_{mm}	/MATS	/(124)	TH3	M	XK1MT
XK2MT		Description not input	/MATS	/(125)	DL2	O	XK2MT
XK3MT		Description not input	/MATS	/(126)	AL7	O	XK3MT
					AL8	O	XK3MT
					AL9	O	XK3MT
XK1MD		Description not input	/MATS	/(127)	TH3	M	XK1MD
XK2MD		Description not input	/MATS	/(128)	AL1	I	XK2MD
					DL2	O	XK2MD
XK3MD		Description not input	/MATS	/(129)	AL7	O	XK3MD
					AL8	O	XK3MD
					AL9	O	XK3MD
XK1MA		Description not input	/MATS	/(130)	TH3	M	XK1MA
XK2MA		Description not input	/MATS	/(131)	AL1	I	XK2MA
					DL2	O	XK2MA
XK3MA		Description not input	/MATS	/(132)	AL4	O	XK3MA
					AL5	M	XK3MA
					AL6	O	XK3MA
					AL7	O	XK3MA
					AL8	O	XK3MA
					AL9	O	XK3MA
XK1ZT	$K^{(1)}_{\tau T}$	The first entry in a 3x3 matrix containing $K_{\tau m}$	/MATS	/(133)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XK1VV	$K^{(I)}_{vv}$	The first entry in a 3x8 matrix containing K_{yv}	/MATS	/(142)	ALGCON	I	XKVV
					TH3	M	XK1VV
					TH4	O	XK1VV
XK2VV		Description not input	/MATS	/(143)	DL2	O	XK2VV
XK3VV		Description not input	/MATS	/(144)	AL3	O	XK3VV
					AL4	O	XK3VV
					AL5	M	XK3VV
					AL6	O	XK3VV
					AL7	O	XK3VV
					AL8	O	XK3VV
					AL9	O	XK3VV
XK3GV		Description not input	/MATS	/(147)	AL4	O	XK3GV
					AL7	O	XK3GV
					AL8	O	XK3GV
XK3PV		Description not input	/MATS	/(150)	AL4	O	XK3PV
XK1RV		Description not input	/MATS	/(151)	TH3	M	XK1RV
XK2RV		Description not input	/MATS	/(152)	DL2	O	XK2RV
XK3RV		Description not input	/MATS	/(153)	AL3	O	XK3RV
					AL4	O	XK3RV
					AL5	M	XK3RV
					AL6	O	XK3RV
					AL7	O	XK3RV
					AL8	O	XK3RV
					AL9	O	XK3RV
XK3OV		Description not input	/MATS	/(156)	AL4	O	XK3OV
XK1MV		Description not input	/MATS	/(160)	TH3	M	XK1MV
XK2MV		Description not input	/MATS	/(161)	DL2	O	XK2MV
XK3MV		Description not input	/MATS	/(162)	AL4	O	XK3MV
					AL5	M	XK3MV
					AL6	O	XK3MV
					AL7	O	XK3MV
					AL8	O	XK3MV
XK1GG	$K^{(I)}_{\gamma\gamma}$	The first entry in a 3x7 matrix containing $K_{y\gamma}$	/MATS	/(166)			
XK3GG		Description not input	/MATS	/(168)	AL4	O	XK3GG
					AL7	O	XK3GG
					AL8	O	XK3GG
					AL9	O	XK3GG
XK3PG		Description not input	/MATS	/(171)	AL4	O	XK3PG
					AL7	O	XK3PG
					AL8	O	XK3PG
XK3RG		Description not input	/MATS	/(174)	AL4	O	XK3RG
					AL7	O	XK3RG
					AL8	O	XK3RG
					AL9	O	XK3RG
XK3OG		Description not input	/MATS	/(177)	AL4	O	XK3OG
					AL7	O	XK3OG
					AL8	O	XK3OG
XK3MG		Description not input	/MATS	/(183)	AL4	O	XK3MG
XK1PP	$K^{(I)}_{\psi\psi}$	The first entry in a 3x6 matrix containing $K_{y\psi}$	/MATS	/(187)			
XK3PP		Description not input	/MATS	/(189)	AL4	O	XK3PP
					AL7	O	XK3PP
					AL8	O	XK3PP
					AL9	O	XK3PP
XK3RP		Description not input	/MATS	/(192)	AL4	O	XK3RP
					AL7	O	XK3RP
					AL8	O	XK3RP
					AL9	O	XK3RP
XK3OP		Description not input	/MATS	/(195)	AL4	O	XK3OP
					AL7	O	XK3OP
					AL8	O	XK3OP
XK3MP		Description not input	/MATS	/(201)	AL4	O	XK3MP

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XK1RR	$K^{(1)}_{hh}$	The first entry in a 3x4 matrix containing K_{yh}	/MATS /(	205)	TH2	O	XK1RR
					TH3	M	XK1RR
					TH4	O	XK1RR
XK2RR		Description not input	/MATS /(	206)	DL2	O	XK2RR
XK3RR		Description not input	/MATS /(	207)	AL3	O	XK3RR
					AL4	O	XK3RR
					AL5	M	XK3RR
					AL6	O	XK3RR
					AL7	O	XK3RR
					AL8	O	XK3RR
					AL9	O	XK3RR
XK3OR		Description not input	/MATS /(	210)	AL4	O	XK3OR
					AL7	O	XK3OR
					AL8	O	XK3OR
XK1MR		Description not input	/MATS /(	214)	TH3	M	XK1MR
XK2MR		Description not input	/MATS /(	215)	DL2	O	XK2MR
XK3MR		Description not input	/MATS /(	216)	AL4	O	XK3MR
					AL5	M	XK3MR
					AL6	O	XK3MR
					AL7	O	XK3MR
					AL8	O	XK3MR
XK1OO	$K^{(1)}_{\rho\rho}$	The first entry in a 3x4 matrix containing $K_{y\rho}$	/MATS /(	220)			
XK3OO		Description not input	/MATS /(	222)	AL4	O	XK3OO
					AL7	O	XK3OO
					AL8	O	XK3OO
					AL9	O	XK3OO
XK3MO		Description not input	/MATS /(	228)	AL4	O	XK3MO
XK1UU	$K^{(1)}_{\mu\mu}$	The first entry in a 3x3 matrix containing $K_{y\mu}$	/MATS /(	232)			
XK1MM	$K^{(1)}_{mm}$	The first entry in a 3x2 matrix containing K_{ym}	/MATS /(	241)	TH3	M	XK1MM
XK2MM		Description not input	/MATS /(	242)	DL2	O	XK2MM
XK3MM		Description not input	/MATS /(	243)	AL4	O	XK3MM
					AL5	M	XK3MM
					AL6	O	XK3MM
					AL7	O	XK3MM
					AL8	O	XK3MM
					AL9	O	XK3MM
XK1ZZ	$K^{(1)}_{\tau\tau}$	The first entry in a 3x1 matrix containing $K_{y\tau}$	/MATS /(	247)	TH2	O	XK1ZZ
XKPI11		The first entry in the 3x3 matrix $-K^{-1}_{m}$	/MATS /(	250)	ALGCON	M	XKPI11
XKPI21		Description not input	/MATS /(	251)	ALGCON	O	XKPI21
XKPI31		Description not input	/MATS /(	252)	ALGCON	O	XKPI31
XKPI12		Description not input	/MATS /(	253)	ALGCON	O	XKPI12
XKPI22		Description not input	/MATS /(	254)	ALGCON	O	XKPI22
XKPI32		Description not input	/MATS /(	255)	ALGCON	O	XKPI32
XKPI13		Description not input	/MATS /(	256)	ALGCON	O	XKPI13
XKPI23		Description not input	/MATS /(	257)	ALGCON	O	XKPI23
XKPI33		Description not input	/MATS /(	258)	ALGCON	O	XKPI33
PA1		Not used	/MATS /(	259)	ALGCON	O	PA1
PA2		$-K^{(2)}_{m}/K^{(2)}_{\delta_E}$	/MATS /(	260)	ALGCON	O	PA2
					AL1	M	PA2
DPDY	$\partial m/\partial y$	A 3x8 matrix that contains the total first partial derivatives of the in-plane-control vector wrt the QL state.	/MATS /(	261)	ALGCON	I	DPDY
					APPLY	I	DPDY
DEPDEY	$\delta p/\delta y$	A 2x8 matrix that contains $\delta p/\delta y = \partial p/\partial y\vert_{m} = \text{constant}$	/MATS /(	285)	ALGCON	M	DEPDEY
					APPLY	I	DEPDEY

824

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DPDL	$\partial m/\partial\lambda$	A 3x3 matrix that contains $\partial m/\partial\lambda_\nu$, $\partial m/\partial\lambda_\gamma$ and $\partial m/\partial\lambda_\psi$	/MATS	/(301)	ALGCON APPLY	I I	DPDL DPDL
PROD2	$\partial(\delta p/\delta y)/\partial y$	A 2x64 matrix that contains the total first partials of the matrix DEPDEY wrt the QL state.	/MATS	/(310)	ALGCON ALGCON APPLY APPLY	I I I I	PROD1 PROD5 PROD1 PROD5
PROD5	$\partial(\partial m/\partial y)/\partial y$	A 3x64 matrix that contains the total first partials of the matrix DPDY wrt the QL state.	/MATS	/(310)			
PROD9		A 2x64 matrix that contains $\partial(\delta p/\delta y)/\partial\lambda_\nu$, $\partial(\delta p/\delta y)/\partial\lambda_\gamma$ and $\partial(\delta p/\delta y)/\partial\lambda_\psi$.	/MATS	/(502)	ALGCON APPLY	I I	PROD9 PROD9
PV	ϕ_ν	See symbol	/MATS	/(550)	AL4	M	PV
PG	ϕ_γ	See symbol	/MATS	/(551)	AL4 APPLY ARCIN CONTRL	M I O O	PG PG PG PG
PP	ϕ_ψ	See symbol	/MATS	/(552)	AL4	M	PP
PR	ϕ_h	See symbol	/MATS	/(553)	AL4	M	PR
PO	ϕ_ρ	See symbol	/MATS	/(554)	AL4	M	PO
PVV	$\phi_{\nu\nu}$	See symbol	/MATS	/(555)	AL4	M	PVV
PGV	$\phi_{\gamma\nu}$	See symbol	/MATS	/(556)	AL4	M	PGV
PPV	$\phi_{\psi\nu}$	See symbol	/MATS	/(557)	AL4	M	PPV
PRV	$\phi_{h\nu}$	See symbol	/MATS	/(558)	AL4	M	PRV
POV	$\phi_{\rho\nu}$	See symbol	/MATS	/(559)	AL4	M	POV
PGG	$\phi_{\gamma\gamma}$	See symbol	/MATS	/(560)	AL4	M	PGG
PPG	$\phi_{\psi\gamma}$	See symbol	/MATS	/(561)	AL4	M	PPG
PRG	$\phi_{h\gamma}$	See symbol	/MATS	/(562)	AL4	M	PRG
POG	$\phi_{\rho\gamma}$	See symbol	/MATS	/(563)	AL4	M	POG
PPP	$\phi_{\psi\psi}$	See symbol	/MATS	/(564)	AL4	M	PPP
PRP	$\phi_{h\psi}$	See symbol	/MATS	/(565)	AL4	M	PRP
POP	$\phi_{\rho\psi}$	See symbol	/MATS	/(566)	AL4	M	POP
PRR	$\phi_{\lambda hh}$	See symbol	/MATS	/(567)	AL4	M	PRR
POR	$\phi_{\rho h}$	See symbol	/MATS	/(568)	AL4	M	POR
POO	$\phi_{\rho\rho}$	See symbol	/MATS	/(569)	AL4	M	POO
PLG	ϕ_{λ_γ}	See symbol	/MATS	/(570)	APPLY ARCIN CONTRL	I O O	PLG PLG PLG
PLP	ϕ_{λ_ψ}	See symbol	/MATS	/(571)	APPLY ARCIN CONTRL	I O O	PLP PLP PLP

325

BLØCK
ØRBIT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
VI	V_I	Inertial velocity	(FT/SEC)	/ORBIT /(	1)	OUTPUT PDBCOL PDBCOL	I I M	VI ORBPRM VI
ORBPRM		Array of intertial and orbital boundary conditions		/ORBIT /(	1)			
GAMI	γ_I	Inertial flight path angles	(RAD)	/ORBIT /(	2)	OUTPUT PDBCOL	I O	GAMI GAMI
PSII	ψ_I	Inertial azimuth	(RAD)	/ORBIT /(	3)	OUTPUT PDBCOL	I M	PSII PSII
XMUI	μ_I	Inertial longitude	(RAD)	/ORBIT /(	4)	OUTPUT PDBCOL	I M	XMUI XMUI
P	P_r	Semi-latus rectum	(FT)	/ORBIT /(	5)	OUTPUT PDBCOL	I M	P P
ECC	e	Orbital eccentricity		/ORBIT /(	6)	OUTPUT PDBCOL	I M	ECC ECC
AIMCL	i	Orbital inclination	(RAD)	/ORBIT /(	7)	OUTPUT PDBCOL	I M	AIMCL AIMCL
ARGP	ω_p	Orbital argument of perigee	(RAD)	/ORBIT /(	8)	OUTPUT PDBCOL	I M	ARGP ARGP
AXCNOD	Ω	Longitude of ascending node	(RAD)	/ORBIT /(	9)	OUTPUT PDBCOL	I M	ASCNOD ASCNOD
SMIMAJ	a_s	Semi-major axis	(FT)	/ORBIT /(	10)	OUTPUT PDBCOL	I M	SMIMAJ SMIMAJ
APOGEE	R_a	Apogee radius	(FT)	/ORBIT /(	11)	OUTPUT PDBCOL	I O	APOGEE APOGEE
PERGEE	R_p	Perigee radius	(FT)	/ORBIT /(	12)	OUTPUT PDBCOL	I O	PERGEE PERGEE
ANOMLY	ζ	True anomoly	(RAD)	/ORBIT /(	13)	OUTPUT PDBCOL	I M	ANOMLY ANOMLY
CAPX	X	Asymptote parameter	T	/ORBIT /(	14)	PDBCOL	O	CAPX
CAPY	Y	Asymptote parameter	(FT)	/ORBIT /(	15)	PDBCOL	O	CAPY
ASYMP	θ	Outgoing asymptote	(RAD)	/ORBIT /(	16)	PDBCOL	M	ASYMP
ENERGY	E	Energy		/ORBIT /(	17)	OUTPUT PDBCOL	I O	ENERGY ENERGY
HMNTM	H	Momentum		/ORBIT /(	18)	OUTPUT PDBCOL	I M	HMNTM HMNTM
DVIDV		Partial derivative of boundary condition		/ORBIT /(	19)	PDBCOL PDBCOL	M I	DVIDV PPO
PPO		Matrix of boundary condition partial derivatives		/ORBIT /(	19)			
DVIDG		Partial derivative of boundary condition		/ORBIT /(	20)	PDBCOL	M	DVIDG
DVIDH		Partial derivative of boundary condition		/ORBIT /(	21)	PDBCOL	M	DVIDH
DVIDM		Partial derivative of boundary condition		/ORBIT /(	22)	PDBCOL	O	DVIDM
DVIDPS		Partial derivative of boundary condition		/ORBIT /(	23)	PDBCOL	M	DVIDPS
DVIDRO		Partial derivative of boundary condition		/ORBIT /(	24)	PDBCOL	M	DVIDRO
DVIDMU		Partial derivative of boundary condition		/ORBIT /(	25)	PDBCOL	O	DVIDMU
DGIDV		Partial derivative of boundary condition		/ORBIT /(	26)	PDBCOL	M	DGIDV
DGIDG		Partial derivative of boundary condition		/ORBIT /(	27)	PDBCOL	M	DGIDG
DGIDH		Partial derivative of boundary condition		/ORBIT /(	28)	PDBCOL	M	DGIDH
DGIDM		Partial derivative of boundary condition		/ORBIT /(	29)	PDBCOL	O	DGIDM
DGIDPS		Partial derivative of boundary condition		/ORBIT /(	30)	PDBCOL	M	DGIDPS
DGIDRO		Partial derivative of boundary condition		/ORBIT /(	31)	PDBCOL	M	DGIDRO
DGIDMU		Partial derivative of boundary condition		/ORBIT /(	32)	PDBCOL	O	DGIDMU
DPIDV		Partial derivative of boundary condition		/ORBIT /(	33)	PDBCOL	M	DPIDV
DPIDG		Partial derivative of boundary condition		/ORBIT /(	34)	PDBCOL	M	DPIDG

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DPIDH		Partial derivative of boundary condition	/ORBIT /(	35)	PDBCQL	M	DPIDH
DPIDM		Partial derivative of boundary condition	/ORBIT /(	36)	PDBCQL	O	DPIDM
DPIDPS		Partial derivative of boundary condition	/ORBIT /(	37)	PDBCQL	M	DPIDPS
DPIDRO		Partial derivative of boundary condition	/ORBIT /(	38)	PDBCQL	M	DPIDRO
DPIDMU		Partial derivative of boundary condition	/ORBIT /(	39)	PDBCQL	O	DPIDMU
DMIDV		Partial derivative of boundary condition	/ORBIT /(	40)	PDBCQL	O	DMIDV
DMIDG		Partial derivative of boundary condition	/ORBIT /(	41)	PDBCQL	O	DMIDG
DMIDH		Partial derivative of boundary condition	/ORBIT /(	42)	PDBCQL	O	DMIDH
DMIDM		Partial derivative of boundary condition	/ORBIT /(	43)	PDBCQL	O	DMIDM
DMIDPS		Partial derivative of boundary condition	/ORBIT /(	44)	PDBCQL	O	DMIDPS
DMIDRO		Partial derivative of boundary condition	/ORBIT /(	45)	PDBCQL	O	DMIDRO
DMIDMU		Partial derivative of boundary condition	/ORBIT /(	46)	PDBCQL	O	DMIDMU
DPDV		Partial derivative of boundary condition	/ORBIT /(	47)	PDBCQL	M	DPDV
DPDG		Partial derivative of boundary condition	/ORBIT /(	48)	PDBCQL	M	DPDG
DPDH		Partial derivative of boundary condition	/ORBIT /(	49)	PDBCQL	M	DPDH
DPDM		Partial derivative of boundary condition	/ORBIT /(	50)	PDBCQL	O	DPDM
DPDPS		Partial derivative of boundary condition	/ORBIT /(	51)	PDBCQL	M	DPDPS
DPDRO		Partial derivative of boundary condition	/ORBIT /(	52)	PDBCQL	M	DPDRO
DPDMU		Partial derivative of boundary condition	/ORBIT /(	53)	PDBCQL	O	DPDMU
DECDV		Partial derivative of boundary condition	/ORBIT /(	54)	PDBCQL	M	DECDV
DECDG		Partial derivative of boundary condition	/ORBIT /(	55)	PDBCQL	M	DECDG
DECDH		Partial derivative of boundary condition	/ORBIT /(	56)	PDBCQL	M	DECDH
DECDM		Partial derivative of boundary condition	/ORBIT /(	57)	PDBCQL	O	DECDM
DECDPS		Partial derivative of boundary condition	/ORBIT /(	58)	PDBCQL	M	DECDPS
DECDRO		Partial derivative of boundary condition	/ORBIT /(	59)	PDBCQL	M	DECDRO
DECDMU		Partial derivative of boundary condition	/ORBIT /(	60)	PDBCQL	O	DECDMU
DIDV		Partial derivative of boundary condition	/ORBIT /(	61)	PDBCQL	M	DIDV
DIDG		Partial derivative of boundary condition	/ORBIT /(	62)	PDBCQL	M	DIDG
DIDH		Partial derivative of boundary condition	/ORBIT /(	63)	PDBCQL	M	DIDH
DIDM		Partial derivative of boundary condition	/ORBIT /(	64)	PDBCQL	O	DIDM
DIDPS		Partial derivative of boundary condition	/ORBIT /(	65)	PDBCQL	M	DIDPS
DIDRO		Partial derivative of boundary condition	/ORBIT /(	66)	PDBCQL	M	DIDRO
DIDMU		Partial derivative of boundary condition	/ORBIT /(	67)	PDBCQL	O	DIDMU
DBEDV		Partial derivative of boundary condition	/ORBIT /(	68)	PDBCQL	O	DBEDV
DBEDG		Partial derivative of boundary condition	/ORBIT /(	69)	PDBCQL	O	DBEDG
DBEDH		Partial derivative of boundary condition	/ORBIT /(	70)	PDBCQL	O	DBEDH
DBEDM		Partial derivative of boundary condition	/ORBIT /(	71)	PDBCQL	O	DBEDM
DBEDPS		Partial derivative of boundary condition	/ORBIT /(	72)	PDBCQL	O	DBEDPS
DBEDRO		Partial derivative of boundary condition	/ORBIT /(	73)	PDBCQL	O	DBEDRO
DBEDMU		Partial derivative of boundary condition	/ORBIT /(	74)	PDBCQL	O	DBEDMU
DNODV		Partial derivative of boundary condition	/ORBIT /(	75)	PDBCQL	O	DNODV
DNODG		Partial derivative of boundary condition	/ORBIT /(	76)	PDBCQL	O	DNODG
DNODH		Partial derivative of boundary condition	/ORBIT /(	77)	PDBCQL	O	DNODH
DNODM		Partial derivative of boundary condition	/ORBIT /(	78)	PDBCQL	O	DNODM
DNODPS		Partial derivative of boundary condition	/ORBIT /(	79)	PDBCQL	O	DNODPS
DNODRO		Partial derivative of boundary condition	/ORBIT /(	80)	PDBCQL	O	DNODRO
DNODMU		Partial derivative of boundary condition	/ORBIT /(	81)	PDBCQL	O	DNODMU
DSMDV		Partial derivative of boundary condition	/ORBIT /(	82)	PDBCQL	M	DSMDV
DSMDG		Partial derivative of boundary condition	/ORBIT /(	83)	PDBCQL	M	DSMDG

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DSMDH		Partial derivative of boundary condition	/ORBIT /(	84)	PDBCOL	M	DSMDH
DSMDM		Partial derivative of boundary condition	/ORBIT /(	85)	PDBCOL	O	DSMDM
DSMDPS		Partial derivative of boundary condition	/ORBIT /(	86)	PDBCOL	M	DSMDPS
DSMDRO		Partial derivative of boundary condition	/ORBIT /(	87)	PDBCOL	M	DSMDRO
DSMDMU		Partial derivative of boundary condition	/ORBIT /(	88)	PDBCOL	O	DSMDMU
DAPDV		Partial derivative of boundary condition	/ORBIT /(	89)	PDBCOL	O	DAPDV
DAPDG		Partial derivative of boundary condition	/ORBIT /(	90)	PDBCOL	O	DAPDG
DAPDH		Partial derivative of boundary condition	/ORBIT /(	91)	PDBCOL	O	DAPDH
DAPDM		Partial derivative of boundary condition	/ORBIT /(	92)	PDBCOL	O	DAPDM
DAPDPS		Partial derivative of boundary condition	/ORBIT /(	93)	PDBCOL	O	DAPDPS
DAPDRO		Partial derivative of boundary condition	/ORBIT /(	94)	PDBCOL	O	DAPDRO
DAPDMU		Partial derivative of boundary condition	/ORBIT /(	95)	PDBCOL	O	DAPDMU
DPEDV		Partial derivative of boundary condition	/ORBIT /(	96)	PDBCOL	O	DPEDV
DPEDG		Partial derivative of boundary condition	/ORBIT /(	97)	PDBCOL	O	DPEDG
DPEDH		Partial derivative of boundary condition	/ORBIT /(	98)	PDBCOL	O	DPEDH
DPEDM		Partial derivative of boundary condition	/ORBIT /(	99)	PDBCOL	O	DPEDM
DPEDPS		Partial derivative of boundary condition	/ORBIT /(	100)	PDBCOL	O	DPEDPS
DPEDRO		Partial derivative of boundary condition	/ORBIT /(	101)	PDBCOL	O	DPEDRO
DPEDMU		Partial derivative of boundary condition	/ORBIT /(	102)	PDBCOL	O	DPEDMU
DANDV		Partial derivative of boundary condition	/ORBIT /(	103)	PDBCOL	O	DANDV
DANDG		Partial derivative of boundary condition	/ORBIT /(	104)	PDBCOL	O	DANDG
DANDH		Partial derivative of boundary condition	/ORBIT /(	105)	PDBCOL	O	DANDH
DANDM		Partial derivative of boundary condition	/ORBIT /(	106)	PDBCOL	O	DANDM
DANDPS		Partial derivative of boundary condition	/ORBIT /(	107)	PDBCOL	O	DANDPS
DANDRO		Partial derivative of boundary condition	/ORBIT /(	108)	PDBCOL	O	DANDRO
DANDMU		Partial derivative of boundary condition	/ORBIT /(	109)	PDBCOL	O	DANDMU
DCXDV		Partial derivative of boundary condition	/ORBIT /(	110)	PDBCOL	O	DCXDV
DCXDG		Partial derivative of boundary condition	/ORBIT /(	111)	PDBCOL	O	DCXDG
DCXDH		Partial derivative of boundary condition	/ORBIT /(	112)	PDBCOL	O	DCXDH
DCXDM		Partial derivative of boundary condition	/ORBIT /(	113)	PDBCOL	O	DCXDM
DCXDPS		Partial derivative of boundary condition	/ORBIT /(	114)	PDBCOL	O	DCXDPS
DCXDRO		Partial derivative of boundary condition	/ORBIT /(	115)	PDBCOL	O	DCXDRO
DCXDMU		Partial derivative of boundary condition	/ORBIT /(	116)	PDBCOL	O	DCXDMU
DCYDV		Partial derivative of boundary condition	/ORBIT /(	117)	PDBCOL	O	DCYDV
DCYDG		Partial derivative of boundary condition	/ORBIT /(	118)	PDBCOL	O	DCYDG
DCYDH		Partial derivative of boundary condition	/ORBIT /(	119)	PDBCOL	O	DCYDH
DCYDM		Partial derivative of boundary condition	/ORBIT /(	120)	PDBCOL	O	DCYDM
DCYDPS		Partial derivative of boundary condition	/ORBIT /(	121)	PDBCOL	O	DCYDPS
DCYDRO		Partial derivative of boundary condition	/ORBIT /(	122)	PDBCOL	O	DCYDRO
DCYDMU		Partial derivative of boundary condition	/ORBIT /(	123)	PDBCOL	O	DCYDMU
DASDV		Partial derivative of boundary condition	/ORBIT /(	124)	PDBCOL	O	DASDV
DASDG		Partial derivative of boundary condition	/ORBIT /(	125)	PDBCOL	O	DASDG
DASDH		Partial derivative of boundary condition	/ORBIT /(	126)	PDBCOL	O	DASDH
DASDM		Partial derivative of boundary condition	/ORBIT /(	127)	PDBCOL	O	DASDM
DASDPS		Partial derivative of boundary condition	/ORBIT /(	128)	PDBCOL	O	DASDPS
DASDRO		Partial derivative of boundary condition	/ORBIT /(	129)	PDBCOL	O	DASDRO
DASDMU		Partial derivative of boundary condition	/ORBIT /(	130)	PDBCOL	O	DASDMU
DENDV		Partial derivative of boundary condition	/ORBIT /(	131)	PDBCOL	O	DENDV
DENDG		Partial derivative of boundary condition	/ORBIT /(	132)	PDBCOL	O	DENDG

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DENDH		Partial derivative of boundary condition		/ORBIT /(	133)	PDBCOL	O	DENDH
DENDM		Partial derivative of boundary condition		/ORBIT /(	134)	PDBCOL	O	DENDM
DENDPS		Partial derivative of boundary condition		/ORBIT /(	135)	PDBCOL	O	DENDPS
DENDRO		Partial derivative of boundary condition		/ORBIT /(	136)	PDBCOL	O	DENDRO
DENDMU		Partial derivative of boundary condition		/ORBIT /(	137)	PDBCOL	O	DENDMU
DMODV		Partial derivative of boundary condition		/ORBIT /(	138)	PDBCOL	O	DMODV
DMODG		Partial derivative of boundary condition		/ORBIT /(	139)	PDBCOL	O	DMODG
DMODH		Partial derivative of boundary condition		/ORBIT /(	140)	PDBCOL	O	DMODH
DMODM		Partial derivative of boundary condition		/ORBIT /(	141)	PDBCOL	O	DMODM
DMODPS		Partial derivative of boundary condition		/ORBIT /(	142)	PDBCOL	O	DMODPS
DMODRO		Partial derivative of boundary condition		/ORBIT /(	143)	PDBCOL	O	DMODRO
DMODMU		Partial derivative of boundary condition		/ORBIT /(	144)	PDBCOL	O	DMODMU
YMXRF	ρ_r	Reference longitude	(RAD)	/ORBIT /(	145)	CHECK	O	YMXRF
						PDBCOL	I	YMXRF
SNXLMR	$\sin(\rho-\rho_r)$	Sine of reference latitude		/ORBIT /(	146)	CHECK	O	SNXLMR
						PDBCOL	I	SNXLMR
CSXLMR	$\cos(\rho-\rho_r)$	Cosine of refernece latitude		/ORBIT /(	147)	CHECK	O	CSXLMR
						PDBCOL	I	CSXLMR
SDOWN	S_D	Down range	(FT)	/ORBIT /(	148)	OUTPUT	I	SDOWN
						PDBCOL	O	SDOWN
SCROSS	S_C	Cross range	(FT)	/ORBIT /(	149)	OUTPUT	I	SCROSS
						PDBCOL	O	SCROSS
TD	θ_D	Down range angle	(RAD)	/ORBIT /(	150)	PDBCOL	M	TD
TC	θ_C	Cross range angle	(RAD)	/ORBIT /(	151)	PDBCOL	M	TC
SNPSR	$\sin(\psi_r)$	Sine of reference azimuth		/ORBIT /(	152)	CHECK	O	SNPSR
						PDBCOL	I	SNPSR
CSPSR	$\cos(\psi_r)$	Cosine of reference azimuth		/ORBIT /(	153)	CHECK	O	CSPSR
						PDBCOL	I	CSPSR
SNGI	$\sin(\gamma_I)$	Sine of inertial flight path angle		/ORBIT /(	154)	PDBCOL	M	SNGI
CSGI	$\cos(\gamma_I)$	Cosine of inertial flight path angle		/ORBIT /(	155)	PDBCOL	M	CSGI
SPSII	$\sin(\psi_I)$	Sin of inertial azimuth		/ORBIT /(	156)	PDBCOL	M	SPSII
CSPSII	$\cos(\psi_I)$	Cosine of inertial azimuth		/ORBIT /(	157)	PDBCOL	M	CPSII
STOT	S_T	Total range	(FT)	/ORBIT /(	158)	OUTPUT	I	STOT
						PDBCOL	O	STOT
						QLTOSZ	I	STOT
CSI	$\cos(i)$	Cosine of inclination		/ORBIT /(	159)	PDBCOL	M	CSI
SNI	$\sin(i)$	Sine of inclination		/ORBIT /(	160)	PDBCOL	M	SNI
SNGNU	$\sin(\nu)$	See symbol		/ORBIT /(	161)	PDBCOL	M	SNGNU
CSANO	$\cos(\zeta)$	See symbol		/ORBIT /(	162)	PDBCOL	M	CSANO
COSDMU	$\cos(\mu-\mu_r)$	See symbol		/ORBIT /(	163)	PDBCOL	M	COSDMU
SINDMU	$\sin(\mu-\mu_r)$	See symbol		/ORBIT /(	164)	PDBCOL	M	SINDMU
THT	θ_T	Total range angle	(FT)	/ORBIT /(	165)	PDBCOL	M	THT
WTFUEL		Description not input		/ORBIT /(	166)	PDBCOL	M	WTFUEL

BLØCK
PC

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
PC1		Not used	/PC	/(1)	GROPE	D	PC
N		Total number of QL state and costate variables. N = 18.	/PC	/(2)	BNDRY	I	N
					CHECK	I	N
					INARC	I	N
					LINDRV	I	N
					NLDRV	I	N
					NOMNAL	I	N
					RKUTT1	I	N
					SALVE	I	N
					WRAPUP	I	N
PC3		Not used	/PC	/(3)			
IDP		Component number that corresponds to the QL state variable ; IDP = 8.	/PC	/(4)	INARC	I	IDP
					WRAPUP	I	IDP
PC5		Not used	/PC	/(5)			
PC6		Not used	/PC	/(6)			
PC7		Not used	/PC	/(7)			
MAXBC		Maximum number of target conditions that QL can handle. MAXBC = 40	/PC	/(8)			
NAUX		Number of velocity loss quantities to be integrated on converged trajectory. NAUX = 5.	/PC	/(9)	WRAPUP	I	NAUX

332

BLØCK
PRINT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AP		Description not input	/PRINT /(	1)	OUTPUT	M	AP
					OLTOSZ	I	AP

BLØCK
S

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
S		An 820 word array used to store the particular and homogeneous solutions.	/S	/(1)	NOMNAL	M	S
					RKUTT1	M	S

BLØCK SIZING

837

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
QP		Description not input	/SIZING/(	5)	QLTOSZ	O	QP
PZ		Description not input	/SIZING/(	20)	QLTOSZ	O	PZ
SW		A synthesis array (20) containing counters and sizing options	/SIZING/(	26)	QLTOSZ	O	SW
SV		A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRQ QLTOSZ	M M	SV SV
SQ		A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRQ QLTOSZ	M M	SQ SQ
SE		Array of synthesis iteration propulsion parameters	/SIZING/(	259)			
QLIM		Saved value of maximum dynamic pressure.	/SIZING/(	264)	ENVPRQ	M	QLIM
WBO		Booster burnout weight (lb)	/SIZING/(	272)			
WLOO		Previous iteration value of booster liftoff weight (lb)	/SIZING/(	273)			
DWEB		Sensitivity of booster stage weight to propellant weight (lb/lb)	/SIZING/(	274)			
DWEO		Sensitivity of orbiter stage weight to propellant weight (lb/lb)	/SIZING/(	275)			
TOLWT		Booster liftoff weight sizing tolerance (lb)	/SIZING/(	276)			
WPB		Booster propellant weight (lb)	/SIZING/(	277)			
TWRAT2		Second stOge thrust-to-weight ratio	/SIZING/(	278)			
BK1		Value of constant weight in booster stage weight equation	/SIZING/(	279)			
BK2		Value of linear term coefficient in booster stage weight equation	/SIZING/(	280)			
BK3		Value of 1/3-power term coefficient in booster stage weight equation	/SIZING/(	281)			
BK4		Value of 2/3-power term coefficient in booster stage weight equation	/SIZING/(	282)			
ISIZE		Sizing option flag 1. Fixed wlo, maximize xpl 2. Fixed xpl, minimize wlo 3. Fixed orbiter, minimize wlo 4. Fixed booster, minimize wlo 5. Fixed (t/w)1.0. Maximize xpl 6. Fixed (t/w)1.0. Determine f	/SIZING/(	283)			
TRAFLG		Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)			
TWRATO		Liftoff thrust-to-weight ratio	/SIZING/(	285)			
OK1		Same as bk1 except for orbiter	/SIZING/(	286)			
OK2		Same as bk2 except for orbiter	/SIZING/(	287)			
OK3		Same as bk3 except for orbiter	/SIZING/(	288)			
OK4		Same as bk4 except for orbiter	/SIZING/(	289)			
PRFLG		Sizing data print flag 1. Print header 2. Print identifier 3. Print data	/SIZING/(	290)			
IPASS		Sizing iteration counter	/SIZING/(	291)			
IPSMAX		Maximum number of iterations	/SIZING/(	292)			
AEXIT		Booster engine exit area (ft**2)	/SIZING/(	293)			
TVACO		Orbiter vacuum thrust (lb)	/SIZING/(	294)			
NO		Number of orbiter engines	/SIZING/(	295)			
WFO		Orbiter burnout weight (lb)	/SIZING/(	296)	QLTOSZ	O	WFO
IDVEL		Total ideal velocity required to orbit (fps)	/SIZING/(	297)	QLTOSZ	O	IDVEL
ISPO		Orbiter vacuum specific impulse sec	/SIZING/(	298)			
ISPB		Booster vacuum specific impulse sec	/SIZING/(	299)			
XPL		Payload weight (lb)	/SIZING/(	300)			
TVACB		Booster vacuum thrust per engine lb	/SIZING/(	301)			
NNB		Number of booster engines	/SIZING/(	302)			

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WEO		Orbiter stage weight (lb)	/SIZING/(	303)			
WEB		Booster stage weight (lb)	/SIZING/(	304)			
WO		Initial orbiter weight (lb)	/SIZING/(	305)			
WLO		Booster liftoff weight (lb)	/SIZING/(	306)			
DVO		Orbiter ideal velocity (fps)	/SIZING/(	307)	QLTOSZ	O	DVO
DVB		Booster idel velocity (fps)	/SIZING/(	308)			
MUB		Booster mass ratio or velocity	/SIZING/(	309)			
MUO		Orbiter mass ratio	/SIZING/(	310)			
VSTG		Booster staging velocity (fps)	/SIZING/(	311)	QLTOSZ	M	VSTG
WPO		Orbiter propellant weight (lb)	/SIZING/(	312)			
JTYP		Sizing. Flag.	/SIZING/(	313)	QLTOSZ WRAPUP	I I	JTYP JTYP
BECO		Booster cut-off arc	/SIZING/(	314)			
BSTG		Booster staging arc	/SIZING/(	315)			
ORBI		Orbiter ignition arc	/SIZING/(	316)	QLTOSZ	I	ORBI
ITNBW		Booster empty weght curve no.	/SIZING/(	317)			
ITNOW		Orbiter empty weight curve no.	/SIZING/(	318)			
SVDPSQ		Saved control metric	/SIZING/(	319)			
SVDCON		Saved payoff improvement	/SIZING/(	320)			
IHUNT		Number of iterations for parameter hunt	/SIZING/(	321)			
SOCO		Solid engine cut-off arc	/SIZING/(	326)			
SOSP		Solid engine drop arc	/SIZING/(	327)			

BLØCK
STUFF

340

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
FK		An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-2 thru -5 in Vol.I of this document.	/STUFF /(	1)	MADAMS RKUTT1 RKUTT1	M I O	F1 FK FS
F2		Description not input	/STUFF /(	821)	MADAMS	I	F2
YP		Description not input	/STUFF /(	1641)	MADAMS	M	YP
YC		Description not input	/STUFF /(	2461)	MADAMS	M	YC

BLØCK
TABLE

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR CODE	SUBROUTINE USAGE VAR
TABLE		A 2100 word array used for storing up to 50 spline fitted univariant tables.	/TABLE /(	1)	SPLINE I SPLINE I	LOCI X
LOCI		A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the initial value of the independent variable of the corresponding table. A zero entry indicates table not input.	/TABLE /(	1)		
X		A 650 word array that contains the independent variable entries of all of the 50 or less tables input.	/TABLE /(	51)		
LOCL		A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that indicates the last interval in which interpolation of the corresponding table occured.	/TABLE /(	701)	SPLINE M SPLINE I	LOCL Y
Y		A 650 word array that contains the dependent variable entries of all of the 50 or less tables input.	/TABLE /(	751)		
LOCF		A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the last value of the independent variable of the corresponding table.	/TABLE /(	1401)	SPLINE I SPLINE I	LOCF Z
Z		The mesh point second derivatives of the cubic spline functions of all of the 50 or less tables input.	/TABLE /(	1451)		

BLØCK
Y

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
Y		An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y	/(1)	GROPE	O	Y
					INARC	M	Y
					MADAMS	M	Y
					QLTOSZ	I	Y
					RKUTT1	M	Y
					SALVE	M	Y
					WRAPUP	I	Y

BLØCK
Z

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
Z	Z	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(	1) BNDRV	I	Z
					BRANPT	I	Z
					ENDPT	I	Z
					ENVPRQ	I	Z
					INTERP	O	Z
					INTRPT	I	Z
					LINDRV	I	Z
					NOMNAL	M	Z
					OUTPUT	I	Z
					RKUTT1	O	Z
					RKUTT2	M	Z
					SALVE	M	Z
					WRAPUP	M	Z

847

BLØCK
ZD

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ZD		A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.I of this document.	/ZD	/(1)	ENVPRO	I	ZD
					LINDRV	I	ZD
					OUTPUT	I	ZD
					RKUTT2	I	ZD
					WRAPUP	I	ZD

849

BLØCK
ZI

850

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ZI		A 20x4 array containing the first four values of Z in the present subarc.	/ZI	/(1)	INTERP	I	ZI
					RKUTTI	I	ZI
					SALVE	M	ZI

10 NOV 72 6.01-47

851

SUBRØUTINE AERØCØ

852

Purpose

AEROCO computes either the univariant or bivariant aerodynamic coefficients C_L and C_D. In addition, it computes the first and second partial derivatives

$$\frac{\partial C_L}{\partial \alpha}, \frac{\partial C_L}{\partial M}, \frac{\partial^2 C_L}{\partial \alpha^2}, \frac{\partial^2 C_L}{\partial M^2} \text{ and } \frac{\partial^2 C_L}{\partial M \partial \alpha}$$

and

$$\frac{\partial C_D}{\partial \alpha}, \frac{\partial C_D}{\partial M}, \frac{\partial^2 C_D}{\partial \alpha^2}, \frac{\partial^2 C_D}{\partial M^2} \text{ and } \frac{\partial^2 C_D}{\partial M \partial \alpha}$$

```
      SUBROUTINE AEROCO                                                 AEROCO
C                                                                       AEROCO
C                    THIS ROUTINE COMPUTES THE VEHICLE*S TOTAL LIFT AND AEROCO
C                    DRAG COEFFICIENTS                                  AEROCO
C                                                                       AEROCO
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF     ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
     *IATM     ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
     *XLMAX    ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB     ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT       ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      , ARCDAT
     *MDB      ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF     ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYNG11,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DATA DEG/57.2957795130823/                                        AEROCO
C                    TEST FOR VARIANCE OF AERO DATA                     AEROCO
      IF(JAER .EQ. 2) GO TO 50                                          AEROCO  50
C                    UNIVARIANT                                         AEROCO
      CL = CLO + ALPHA*CLA                                              AEROCO
      CD = CDO + FK*CL**2                                               AEROCO
      CDA = 2.*FK*CL*CLA                                                AEROCO
      CLM = CLOM + ALPHA*CLAM                                           AEROCO
      CDM = CDOM + CL*(FKM*CL + 2.*FK*CLM)                              AEROCO
      CLAA = 0.                                                         AEROCO
      CDAA = 2.*FK*CLA**2                                               AEROCO
      CLMM = CLOMM + ALPHA*CLAMM                                        AEROCO
      CDMM = CDOMM + 2.*CLM*(FKM*CL + FK*CLM) + CL*(FKMM*CL + 2.*(FKM    AEROCO
     **CLM + FK*CLMM))                                                  AEROCO
      CDAM = 2.*(FK*(CL*CLAM + CLM*CLA) + FKM*CL*CLA)                   AEROCO
      RETURN                                                            AEROCO
C                    BIVARIANT                                          AEROCO
   50 CALL BLINE(ALPHA*DEG, MACH, CL)                                   AEROCO
      CLA = CLA*DEG                                                     AEROCO
      CLAM = CLAM*DEG                                                   AEROCO
      CLAA = CLAA*DEG**2                                                AEROCO
      CDA = CDA*DEG                                                     AEROCO
      CDAM = CDAM*DEG                                                   AEROCO
      CDAA = CDAA*DEG**2                                                AEROCO
      RETURN                                                            AEROCO
```

76. END

AEROCO

6 OCT 72 6.01-44

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ALPHA	α	I	Angle of attack	(RAD)	/DYNA	/(79)	AEROCO	I	ALPHA
							ALGCON	M	ALPHA
							AL2	I	ALPHA
							ARCIN	M	ALPHA
							CONTRL	M	ALPHA
							ENVPRO	I	ALPHA
							MBMECO	I	ALPHA
							NPLANE	I	ALPHA
							OUTPUT	I	ALPHA
							TRAJIN	O	ALPHA
							UT	I	ALPHA
							WRAPUP	I	ALPHA
CD	C_D	O	Drag coefficient		/DYNA	/(192)	AEROCO	O	CD
							OUTPUT	I	CD
							UT	I	CD
CDA	C_{D_α}	M	Drag coefficient slope	(RAD^{-1})	/DYNA	/(193)	AEROCO	M	CDA
							UT	I	CDA
CDAA	$\partial C_{D_\alpha}/\partial\alpha$	M	See symbol		/DYNA	/(195)	AEROCO	M	CDAA
							UT	I	CDAA
CDAM	$\partial C_{D_\alpha}/\partial M$	M	See symbol		/DYNA	/(197)	AEROCO	M	CDAM
							UT	I	CDAM
CDM	$\partial C_D/\partial M$	O	See symbol		/DYNA	/(194)	AEROCO	O	CDM
							UT	I	CDM
CDMM	$\partial^2 C_D/\partial M^2$	O	See symbol		/DYNA	/(196)	AEROCO	O	CDMM
							UT	I	CDMM
CDO	C_{D_0}	I	Drag coefficient at $\alpha = 0$		/DYNA	/(104)	AEROCO	I	CDO
							STATEF	I	CDO
CDOM	$\partial C_{D_0}/\partial M$	I	See symbol		/DYNA	/(105)	AEROCO	I	CDOM
							STATEF	I	CDOM
CDOMM	$\partial^2 C_{D_0}/\partial M^2$	I	See symbol		/DYNA	/(145)	AEROCO	I	CDOMM
							STATEF	I	CDOMM
CL	C_L	M	Lift coefficient		/DYNA	/(186)	AEROCO	M	CL
							OUTPUT	I	CL
							UT	I	CL
CLA	C_{L_α}	M	Lift coefficient slope	(RAD^{-1})	/DYNA	/(187)	AEROCO	M	CLA
							STATEF	M	CLA
							UT	I	CLA
CLAA	$\partial C_{L_\alpha}/\partial\alpha$	M	See symbol		/DYNA	/(189)	AEROCO	M	CLAA
							UT	I	CLAA
CLAM	$\partial C_{L_\alpha}/\partial M$	M	See symbol		/DYNA	/(191)	AEROCO	M	CLAM
							STATEF	M	CLAM
							UT	I	CLAM
CLAMM	$\partial^2 C_{L_\alpha}/\partial M^2$	I	See symbol		/DYNA	/(146)	AEROCO	I	CLAMM
							STATEF	M	CLAMM
CLM	$\partial C_L/\partial M$	M	See symbol		/DYNA	/(188)	AEROCO	M	CLM
							UT	I	CLM
CLMM	$\partial^2 C_L/\partial M^2$	M	See symbol		/DYNA	/(190)	AEROCO	M	CLMM
							UT	I	CLMM
CLO	C_{L_0}	I	Lift coefficient at $\alpha = 0$		/DYNA	/(106)	AEROCO	I	CLO
							STATEF	I	CLO
CLOM	$\partial C_{L_0}/\partial M$	I	See symbol		/DYNA	/(147)	AEROCO	I	CLOM
							STATEF	I	CLOM
CLOMM	$\partial^2 C_{L_0}/\partial M^2$	I	See symbol		/DYNA	/(148)	AEROCO	I	CLOMM
							STATEF	I	CLOMM
FK	k	I	Induced drag coefficient		/DYNA	/(107)	AEROCO	I	FK
							STATEF	I	FK
FKM	$\partial k/\partial M$	I	See symbol		/DYNA	/(182)	AEROCO	I	FKM
							STATEF	I	FKM
FKMM	$\partial^2 k/\partial M^2$	I	See symbol		/DYNA	/(183)	AEROCO	I	FKMM
							STATEF	I	FKMM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
JAER		I	Aerodynamic model option flag	/ARCDAT/(	9)	AERODO	I	JAER
						ARCIN	I	JAER
						OUTPUT	I	JAER
						STATEF	I	JAER
						UT	I	JAER
MACH	M	I	Mach number	/DYNA /(	26)	AERODO	I	MACH
						ENVPRO	I	MACH
						OUTPUT	I	MACH
						STATEF	M	MACH

SUBRØUTINE
ALGCØN

358

Purpose

ALGCON has two purposes. The first is to determine, by means of a Newton-Raphson iteration, a value of w, the in-plane control vector, that causes the vector of algebraic constraints

$$K = (K^{(1)}, K^{(2)}, K^{(3)})^T$$

to vanish. Having accomplished this, ALGCON's second purpose is to determine, by means of the implicit function theorem, the following matrices:

i) $\partial w/\partial y$;

ii) $\delta p/\delta y = \partial p/\partial y|_{\alpha=\text{constant}}$, if α is optimal;

III) if α is nonoptimal

$\partial(\partial w/\partial y)/\partial y_i$, $i=1,2,\ldots,8$;

iv) if α is optimal,

$\partial(\delta p/\delta y)/\partial y_i$, $i=1,2,\ldots,8$;

v) if α is optimal,

$\partial w/\partial \lambda$, where $\lambda=(\lambda_V,\lambda_\gamma,\lambda_\psi)^T$;

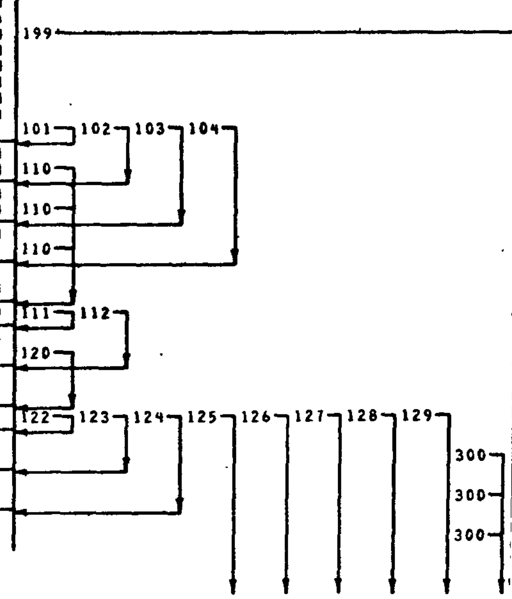
vi) if α is optimal,

$\partial(\delta p/\delta y)/\partial \lambda_i$, $i=1,2,3$.*

*See Sections 9, 10, 16 and 17 of Vol. I.

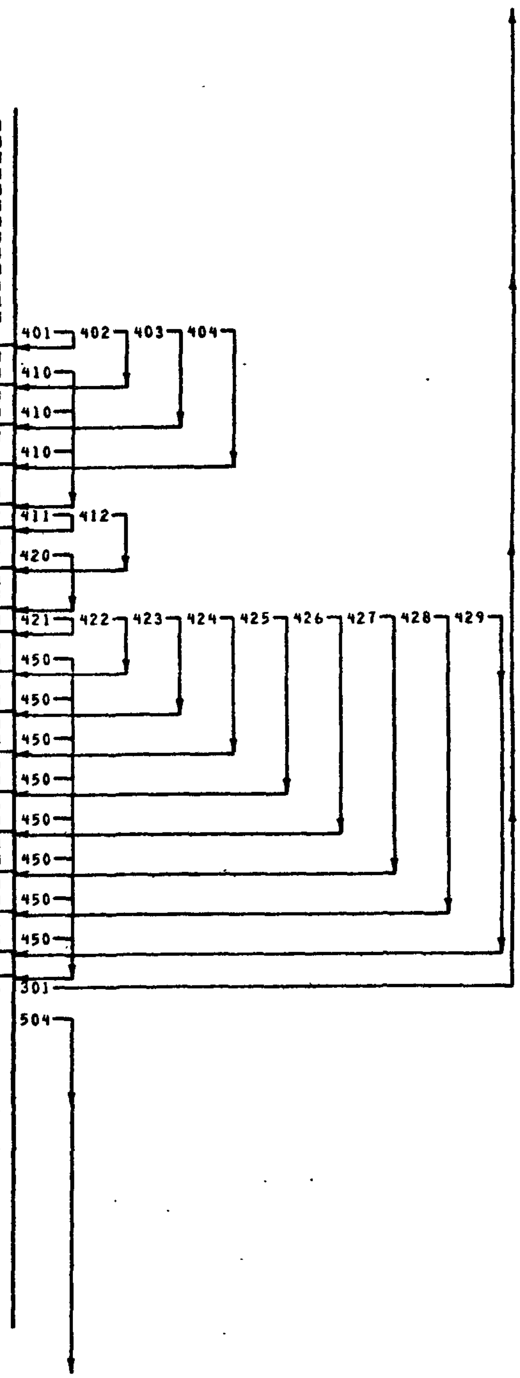
```
      SUBROUTINE ALGCON(JP1, JP2, JP3)                                  ALGCON
C                                                                       ALGCON
C                         THIS ENTRY CONTROLS THE ITERATION FOR THE IN-PLANE   ALGCON
C                         CONTROL. JP1, JP2 AND JP3 POINT TO THE GOVERNING     ALGCON
C                         EQS. FOR THRUST, DEFLECTION AND ANGLE OF ATTACK      ALGCON
      DIMENSION DP(3), P(1), YKPI(2, 2), XKY(3, 1), SKWW(3, 9),         ALGCON
     *XKWW(3, 1), SKYYI(3, 8), XKYY(3, 1), YY(3, 8), ZZ(3, 3),          ALGCON
     *SLWV(2, 6), SLYLI(2, 8), SLYYI(2, 8), XKWY(3, 1), VV(2, 8),       ALGCON
     *UU(2, 2)                                                          ALGCON
      EQUIVALENCE (P, P1), (YKPI, YKPI11), (YKPI21, YKPI11(2)),         ALGCON
     *(YKPI12, YKPI11(3)), (YKPI22, YKPI11(4)), (XKY, XK1V),            ALGCON
     *(SKWW, SLWV), (XKWW, XK1TT), (SKYYI, SLYLI, SLYYI), (XKYY, XK1VV), ALGCON
     *(XKWY, XK1VT), (XKL21, XKL11(2)), (XKL31, XKL11(3)),              ALGCON
     *(XKL12, XKL11(4)), (XKL22, XKL11(5)), (XKL32, XKL11(6)),          ALGCON
     *(XKL13, XKL11(7)), (XKL23, XKL11(8)), (XKL33, XKL11(9)),          ALGCON
     *(YY, VV), (ZZ, UU)                                                ALGCON
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNTC16,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, DV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMA, ISPAT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INSF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBM,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVM                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
```

```
      *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,     MATS
      *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,     MATS
      *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,     MATS
      *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,     MATS
      *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,     MATS
      *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT       MATS
       COMMON /MATS/                                                         MATS
      *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,     MATS
      *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,     MATS
      *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,     MATS
      *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,     MATS
      *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,     MATS
      *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,     MATS
      *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,     MATS
      *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,     MATS
      *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
      *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
      *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
      *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
      *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
      *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
      *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                MATS
       COMMON /MATS/                                                         MATS
      *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)     MATS
       COMMON /MATS/                                                         MATS
      *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,     MATS
      *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,     MATS
      *POR   ,POO   ,PLG   ,PLP                                              MATS
       EQUIVALENCE(PROD1,PROD5)                                              MATS
       DATA XALGCN /6HALGCON/, XKL11, XKL21, XKL12, XKL22, XKL13, XKL23    ALGCON
      */6*0./                                                                ALGCON
 C                     IN-PLANE CONTROL ITERATION                            ALGCON
       ASSIGN 302 TO JSWTCH                                                  ALGCON
       PA1 = 0.                                                              ALGCON
       PA2 = 0.                                                              ALGCON
       DO 10 I = 1, 40                                                       ALGCON
    10 P(I) = 0.                                                             ALGCON
 C                     STORE INITIAL GUESS                                   ALGCON
       P1 = T                                                                ALGCON
       P2 = DELTAE                                                           ALGCON
       P3 = ALPHA                                                            ALGCON
       ITR = 0                                                               ALGCON
 C                     IS THIS OPTIMAL CONTROL                               JUL19A
       IF(JP3 .EQ. 1) GO TO 199                                              ALGCON
 C                     NONOPTIMAL ANGLE OF ATTACK                            ALGCON
       IP3 = JP3 - 1                                                         ALGCON
       ASSIGN 100 TO ISWTCH                                                  ALGCON
   100 CALL UTNOP                                                            ALGCON
 C                     GO TO APPROPRIATE THRUST EQUATION                     ALGCON
       GO TO (101, 102, 103, 104), JP1                                       JUL21
   101 CALL TH1001                                                           ALGCON
       GO TO 110                                                             ALGCON
   102 CALL TH2001                                                           ALGCON
       GO TO 110                                                             ALGCON
   103 CALL TH3001                                                           ALGCON
       GO TO 110                                                             JUL21
   104 CALL TH4001                                                           JUL21
 C                     GO TO APPROPRIATE DEFLECTION EQUATION                 ALGCON
   110 GO TO (111, 112), JP2                                                 ALGCON
   111 CALL DL1001                                                           ALGCON
       GO TO 120                                                             ALGCON
   112 CALL DL2001                                                           ALGCON
 C                     GO TO APPROPRIATE ANGLE OF ATTACK EQUATION            ALGCON
   120 GO TO (122, 123, 124, 125, 126, 127, 128, 129), IP3                   ALGCON
   122 CALL AL2001                                                           ALGCON
       GO TO 300                                                             ALGCON
   123 CALL AL3001                                                           ALGCON
       GO TO 300                                                             ALGCON
   124 CALL AL4001                                                           ALGCON
       GO TO 300                                                             ALGCON
```



```
  125 CALL AL5001                                                  ALGCON
      GO TO 300                                                    ALGCON
  126 CALL AL6001                                                  ALGCON
      GO TO 300                                                    ALGCON
  127 CALL AL7001                                                  ALGCON
      GO TO 300                                                    ALGCON
  128 CALL AL8001                                                  ALGCON
      GO TO 300                                                    ALGCON
  129 CALL AL9001                                                  ALGCON
      GO TO 300                                                    ALGCON
C                       OPTIMAL ANGLE OF ATTACK                    ALGCON
  199 ASSIGN 200 TO ISWTCH                                         ALGCON
  200 CALL UTOP                                                    ALGCON
C                       GO TO APPROPRIATE THRUST EQUATION          ALGCON
      GO TO (201, 202, 203, 204), JP1                              JUL21
  201 CALL TH1002                                                  ALGCON
      GO TO 210                                                    ALGCON
  202 CALL TH2002                                                  ALGCON
      GO TO 210                                                    ALGCON
  203 CALL TH3002                                                  ALGCON
      GO TO 210                                                    JUL21
  204 CALL TH4002                                                  JUL21
C                       GO TO APPROPRIATE DEFLECTION EQUATION      ALGCON
  210 GO TO (211, 212), JP2                                        ALGCON
  211 CALL DL1002                                                  ALGCON
      GO TO 220                                                    ALGCON
  212 CALL DL2002                                                  ALGCON
C                       GO TO OPTIMAL ANGLE OF ATTACK EQUATION     ALGCON
  220 CALL AL1001                                                  ALGCON
  300 ITR = ITR + 1                                                ALGCON
C                       TEST FOR TOO MANY ITERATIONS               ALGCON
      IF(ITR .GT. 20) CALL ERROR(XALGCN, -1, 0)                    ALGCON
  301 CONTINUE                                                     ALGCON
C                       COMPUTE INVERSE OF PARTIALS MATRIX         ALGCON
      DET = XK2T*(XK1D*XK3A - XK1A*XK3D) + XK2D*(XK1A*XK3T - XK1T*XK3A)  ALGCON
     * + XK2A*(XK1T*XK3D - XK1D*XK3T)                              ALGCON
C                       TEST FOR SINGULARITY                       ALGCON
      IF(ABS(DET) .LT. 1.E-14) CALL ERROR(XALGCN, -2, 0)           ALGCON
      XKPI11 = (XK2D*XK3A - XK2A*XK3D)/DET                         ALGCON
      XKPI21 = (XK2A*XK3T - XK2T*XK3A)/DET                         ALGCON
      XKPI31 = (XK2T*XK3D - XK2D*XK3T)/DET                         ALGCON
      XKPI12 = (XK1A*XK3D - XK1D*XK3A)/DET                         ALGCON
      XKPI22 = (XK1T*XK3A - XK1A*XK3T)/DET                         ALGCON
      XKPI32 = (XK1D*XK3T - XK1T*XK3D)/DET                         ALGCON
      XKPI13 = (XK1D*XK2A - XK1A*XK2D)/DET                         ALGCON
      XKPI23 = (XK1A*XK2T - XK1T*XK2A)/DET                         ALGCON
      XKPI33 = (XK1T*XK2D - XK1D*XK2T)/DET                         ALGCON
      GO TO JSWTCH                                                 ALGCON
C                       COMPUTE INCREMENT FOR IN PLANE CONTROL     ALGCON
  302 CALL MATMLT(DP, XKPI11, XK1, 3, 3, 1)                        ALGCON
C                       CHECK FOR CONVERGENCE                      ALGCON
      SUM = 0.                                                     ALGCON
      DO 306 I = 1, 3                                              ALGCON
      DIV = P(I)                                                   ALGCON
      IF(DIV .EQ. 0.) DIV = 1.                                     ALGCON
  306 SUM = SUM + ABS(DP(I)/DIV)                                   ALGCON
      IF(SUM .LE. 1.E-12) GO TO 307                                ALGCON
C                       NOT CONVERGED YET                          ALGCON
      CALL MATADD(P, P, DP, 3, 1)                                  ALGCON
      T = P1                                                       ALGCON
      DELTAE = P2                                                  ALGCON
      ALPHA = P3                                                   ALGCON
      GO TO ISWTCH                                                 ALGCON
C                       CONVERGED                                  ALGCON
  307 CALL MATADD(P, P, DP, 3, 1)                                  ALGCON
      T = P1                                                       ALGCON
      DELTAE = P2                                                  ALGCON
```

```
      ALPHA = P3                                                  ALGCON
      RETURN                                                      ALGCON
C                                                                 ALGCON
C                   THIS ENTRY COMBINES UP THE FIRST AND SECOND PARTIALS ALGCON
C                   OF THE IN PLANE CONTROL WITH RESPECT TO THE STATE    ALGCON
C                   AND COSTATE                                   ALGCON
C                                                                 ALGCON
      ENTRY ALGCNV                                                ALGCON
      ASSIGN 500 TO JSWTCH                                        ALGCON
      DO 399 I = 4, 549                                           ALGCON
  399 P(I) = 0.                                                   ALGCON
  400 CALL UT                                                     ALGCON
C                   GO TO APPROPRIATE THRUST EQUATION             ALGCON
      GO TO (401, 402, 403, 404), JP1                             JUL21
  401 CALL TH1                                                    ALGCON
      GO TO 410                                                   ALGCON
  402 CALL TH2                                                    ALGCON
      GO TO 410                                                   ALGCON
  403 CALL TH3                                                    ALGCON
      GO TO 410                                                   JUL21
  404 CALL TH4                                                    JUL21
C                   GO TO APPROPRIATE DEFLECTION EQUATION         ALGCON
  410 GO TO (411, 412), JP2                                       ALGCON
  411 CALL DL1                                                    ALGCON
      GO TO 420                                                   ALGCON
  412 CALL DL2                                                    ALGCON
C                   GO TO APPROPRIATE ANGLE OF ATTACK EQUATION    ALGCON
  420 GO TO(421, 422, 423, 424, 425, 426, 427, 428, 429), JP3     ALGCON
  421 CALL AL1                                                    ALGCON
      GO TO 450                                                   ALGCON
  422 CALL AL2                                                    ALGCON
      GO TO 450                                                   ALGCON
  423 CALL AL3                                                    ALGCON
      GO TO 450                                                   ALGCON
  424 CALL AL4                                                    ALGCON
      GO TO 450                                                   ALGCON
  425 CALL AL5                                                    ALGCON
      GO TO 450                                                   ALGCON
  426 CALL AL6                                                    ALGCON
      GO TO 450                                                   ALGCON
  427 CALL AL7                                                    ALGCON
      GO TO 450                                                   ALGCON
  428 CALL AL8                                                    ALGCON
      GO TO 450                                                   ALGCON
  429 CALL AL9                                                    ALGCON
  450 GO TO 301                                                   ALGCON
C                   TEST FOR OPTIMAL OR NONOPTIMAL ANGLE OF ATTACK ALGCON
  500 IF(JP3 .GT. 1) GO TO 504                                    ALGCON
C                   OPTIMAL.  COMPUTE PARTIALS OF THRUST AND DEFL. WITH ALGCON
C                   RESPECT TO STATE AS THOUGH ALPHA WERE CONSTANT. ALGCON
      DET = XK1D*XK2T - XK1T*XK2D                                 ALGCON
      YKPI11 = XK2D/DET                                           ALGCON
      YKPI21 =-XK2T/DET                                           ALGCON
      YKPI12 =-XK1D/DET                                           ALGCON
      YKPI22 = XK1T/DET                                           ALGCON
      DO 503 I = 1, 2                                             ALGCON
      DO 502 J = 1, 8                                             ALGCON
      SUM = 0.                                                    ALGCON
      DO 501 K = 1, 2                                             ALGCON
  501 SUM = SUM + YKPI(I, K)*XKY(K, J)                            ALGCON
  502 DEPDEY(I, J) = SUM                                          ALGCON
  503 CONTINUE                                                    ALGCON
C                   IF QL IS CONVERGED WE ARE DONE.               ALGCON
      IF(KONVER) RETURN                                           ALGCON
C                   NONOPTIMAL OR OPTIMAL AND QL NOT CONVERGED.  COMPUTE ALGCON
C                   TOTAL FIRST PARTIALS OF IN PLANE CONTROL WITH ALGCON
```



```
      C                        RESPECT TO STATE.                                 ALGCON
        504 CALL MATMLT(DPDY, XKPI11, XK1V, 3, 3, 8)                             ALGCON
      C                        IF QL IS CONVERGED WE ARE DONE.                   ALGCON
            IF(KONVER) RETURN                                                    ALGCON
      C                        TEST FOR OPTIMAL OR NONOPTIMAL ANGLE OF ATTACK    ALGCON
            IF(JP3 .EQ. 1) GO TO 517                                             ALGCON
      C                        NONOPTIMAL.  COMPUTE TOTAL SECOND PARTIALS OF IN- ALGCON
      C                        PLANE CONTROL WITH RESPECT TO STATE.  RESULTS GO IN ALGCON
      C                        ARRAY PROD5.                                      ALGCON
            KP = 0                                                               ALGCON
            KLIM = 0                                                             ALGCON
            INC = 3                                                              ALGCON
            DO 509 K = 1, 3                                                      ALGCON
            KLIM = KLIM + INC                                                    ALGCON
            INK = 3                                                              ALGCON
            KK = K                                                               JUL21
       5041 KP = KP + 1                                                          JUL21
            DO 505 I = 1, 3                                                      ALGCON
        505 SKWW(I, KP) = XKWW(I, KK)                                            ALGCON
            IF(INK + K - 4) 506, 506, 507                                        ALGCON
        506 INK = 1                                                              ALGCON
            GO TO 508                                                            ALGCON
        507 INK = INK - 1                                                        ALGCON
        508 KK = KK + INK                                                        JUL21
            IF(KK .LE. KLIM) GO TO 5041                                          JUL21
        509 INC = INC - 1                                                        ALGCON
            III = -7                                                             ALGCON
            II = -2                                                              ALGCON
            JLIM = 0                                                             ALGCON
            INC = 8                                                              ALGCON
            DO 516 I = 1, 8                                                      ALGCON
            III = III + 8                                                        ALGCON
            II = II + 3                                                          ALGCON
            JLIM = JLIM + INC                                                    ALGCON
            INK = 8                                                              ALGCON
            J = 0                                                                ALGCON
            JJ = I                                                               JUL21
       5091 J = J + 1                                                            JUL21
            DO 510 K = 1, 3                                                      ALGCON
        510 SKYYI(K, J) = XKYY(K, JJ)                                            ALGCON
            IF(INK + I - 9) 511, 511, 512                                        ALGCON
        511 INK = 1                                                              ALGCON
            GO TO 513                                                            ALGCON
        512 INK = INK - 1                                                        ALGCON
        513 JJ = JJ + INK                                                        JUL21
            IF(JJ .LE. JLIM) GO TO 5091                                          JUL21
            JJ = -2                                                              ALGCON
            DO 514 J = 1, 8                                                      ALGCON
            JJ = JJ + 3                                                          ALGCON
        514 CALL MATMLT(YY(1, J), XKWY(1, JJ), DPDY(1, I), 3, 3, 1)              ALGCON
            CALL MATADD(SKYYI, SKYYI, YY, 3, 8)                                  ALGCON
            KK = - 2                                                             ALGCON
            DO 515 K = 1, 3                                                      ALGCON
            KK = KK + 3                                                          ALGCON
        515 CALL MATMLT(ZZ(1, K), SKWW(1, KK), DPDY(1, I), 3, 3, 1)              ALGCON
            CALL MATADD(ZZ, ZZ, XKWY(1, II), 3, 3)                               ALGCON
            CALL MATMLT(YY, ZZ, DPDY, 3, 3, 8)                                   ALGCON
            CALL MATADD(SKYYI, SKYYI, YY, 3, 8)                                  ALGCON
            CALL MATMLT(PROD5(1, III), XKPI11, SKYYI, 3, 3, 8)                   ALGCON
        516 INC = INC - 1                                                        ALGCON
            RETURN                                                               ALGCON
      C                        OPTIMAL.  COMPUTE TOTAL FIRST PARTIALS OF IN PLANE ALGCON
      C                        CONTROL WITH RESPECT TO COSTATE.                  ALGCON
        517 XKL31 = VDA                                                          ALGCON
            XKL32 = GDA                                                          ALGCON
            XKL33 = PDA                                                          ALGCON
            CALL MATMLT(DPDL, XKPI11, XKL11, 3, 3, 3)                            ALGCON
      C                        COMPUTE TOTAL FIRST PARTIALS OF THE MATRIX DEPDEY ALGCON
      C                        WITH RESPECT TO THE STATE.  DEPDEY WAS COMPUTED   ALGCON
      C                        ABOVE.  RESULTS GO IN ARRAY PROD1.                ALGCON
```

```
      KP = 0                                                        ALGCON
      KLIM = 0                                                      ALGCON
      INC = 3                                                       ALGCON
      DO 522 K = 1, 2                                               ALGCON
      KLIM = KLIM + INC                                             ALGCON
      INK = 3                                                       ALGCON
      KK = K                                                        JUL21
 5171 KP = KP + 1                                                   JUL21
      DO 518 I = 1, 2                                               ALGCON
  518 SLWV(I, KP) = XKWW(I, KK)                                     ALGCON
      IF(INK + K - 4) 519, 519, 520                                 ALGCON  519  520
  519 INK = 1                                                       ALGCON
      GO TO 521                                                     ALGCON  521
  520 INK = INK - 1                                                 ALGCON
  521 KK = KK + INK                                                 JUL21
      IF(KK .LE. KLIM) GO TO 5171                                   JUL21   5171
  522 INC = INC - 1                                                 ALGCON
      II = -7                                                       ALGCON
      DO 524 I = 1, 3                                               ALGCON
      II = II + 8                                                   ALGCON
      JJ = -2                                                       ALGCON
      DO 5223 J = 1, 8                                              ALGCON
      JJ = JJ + 3                                                   ALGCON
      DO 5222 K = 1, 2                                              ALGCON
      JJJ = JJ                                                      ALGCON
      SUM = 0.                                                      ALGCON
      DO 5221 L = 1, 3                                              ALGCON
      SUM = SUM + XKWY(K, JJJ)*DPDL(L, I)                           ALGCON
 5221 JJJ = JJJ + 1                                                 ALGCON
 5222 VV(K, J) = SUM                                                ALGCON
 5223 CONTINUE                                                      ALGCON
      KK = -2                                                       ALGCON
      DO 523 K = 1, 2                                               ALGCON
      KK = KK + 3                                                   ALGCON
  523 CALL MATMLT( UU(1, K), SLWV(1, KK), DPDL(1, I), 2, 3, 1)      ALGCON
      CALL MATMLT(SLYLI, UU, DEPDEY, 2, 2, 8)                       ALGCON
      CALL MATADD(SLYLI, SLYLI, VV, 2, 8)                           ALGCON
  524 CALL MATMLT(PROD9(1, II), YKPI, SLYLI, 2, 2, 8)               ALGCON
      III = -7                                                      ALGCON
      II = -2                                                       ALGCON
      JLIM = 0                                                      ALGCON
      INC = 8                                                       ALGCON
      DO 535 I = 1, 8                                               ALGCON
      III = III + 8                                                 ALGCON
      II = II + 3                                                   ALGCON
      JLIM = JLIM + INC                                             ALGCON
      INK = 8                                                       ALGCON
      J = 0                                                         ALGCON
      JJ = I                                                        JUL21
 5241 J = J + 1                                                     JUL21
      DO 525 K = 1, 2                                               ALGCON
  525 SLYYI(K, J) = XKYY(K, JJ)                                     ALGCON
      IF(INK + I - 9) 526, 526, 527                                 ALGCON  526  527
  526 INK = 1                                                       ALGCON
      GO TO 528                                                     ALGCON  528
  527 INK = INK - 1                                                 ALGCON
  528 JJ = JJ + INK                                                 JUL21
      IF(JJ .LE. JLIM) GO TO 5241                                   JUL21   5241
      JJ = -2                                                       ALGCON
      DO 531 J = 1, 8                                               ALGCON
      JJ = JJ + 3                                                   ALGCON
      DO 530 K = 1, 2                                               ALGCON
      JJJ = JJ                                                      ALGCON
      SUM = 0.                                                      ALGCON
      DO 529 L = 1, 3                                               ALGCON
      SUM = SUM + XKWY(K, JJJ)*DPDY(L, I)                           ALGCON
  529 JJJ = JJJ + 1                                                 ALGCON
  530 VV(K, J) = SUM                                                ALGCON
  531 CONTINUE                                                      ALGCON
      CALL MATADD(SLYYI, SLYYI, VV, 2, 8)                           ALGCON
      KK = -2                                                       ALGCON
```

```
      DO 532 K = 1, 2                                                  ALGCON
      KK = KK + 3                                                      ALGCON
  532 CALL MATMLT(UU(1, K), SLWV(1, KK), DPDY(1, I), 2, 3, 1)          ALGCON
      DO 534 K = 1, 2                                                  ALGCON
      JJ = II                                                          ALGCON
      DO 533 J = 1, 2                                                  ALGCON
      UU(K, J) = UU(K, J) + XKWY(K, JJ)                                ALGCON
  533 JJ = JJ + 1                                                      ALGCON
  534 CONTINUE                                                         ALGCON
      CALL MATMLT(VV, UU, DEPDEY, 2, 2, 8)                             ALGCON
      CALL MATADD(SLYYI, SLYYI, VV, 2, 8)                              ALGCON
      CALL MATMLT(PROD1(1, III), VKPI, SLYYI, 2, 2, 8)                 ALGCON
  535 INC = INC - 1                                                    ALGCON
      RETURN                                                           ALGCON
      END                                                              ALGCON
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	M	Angle of attack	(RAD)	/DYNA	/(79)	AERDCO ALGCON AL2 ARCIN CONTRL ENVPRO MOMECO NPLANE OUTPUT TRAJIN UT WRAPUP	I M I M M I I I I O I I	ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA
DELTAE	δ_E	M	Engine deflection	(RADS)	/DYNA	/(155)	ALGCON ARCIN CONTRL DL1 OUTPUT TRAJIN UT	M M M I I O I	DELTAE DELTAE DELTAE DELTAE DELTAE DELTAE DELTAE
DEPDEY	$\delta p/\delta y$	M	A 2x8 matrix that contains $\delta p/\delta y = \partial p/\partial y\|_w =$ constant		/MATS	/(285)	ALGCON APPLY	M I	DEPDEY DEPDEY
DPDL	$\partial w/\partial\lambda$	I	A 3x3 matrix that contains $\partial w/\partial\lambda_v$, $\partial w/\partial\lambda_\gamma$ and $\partial w/\partial\lambda_\psi$		/MATS	/(301)	ALGCON APPLY	I I	DPDL DPDL
DPDY	$\partial w/\partial y$	I	A 3x8 matrix that contains the total first partial derivatives of the in-plane-control vector wrt the QL state.		/MATS	/(261)	ALGCON APPLY	I I	DPDY DPDY
GDA	$K^{(3)}_{\lambda_\gamma}$	I	Explicit partial of $K^{(3)}$ wrt λ_γ when w is optimal.		/MATS	/(17)	ALGCON AL1	I O	GDA GDA
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.		/CNTRL	/(28)	ALGCON APPLY ARCIN COHOMO GROPE NLDRV OUTPUT RKUTT1	I I I O O I I I	KONVER KONVER KONVER KONVER KONVER KONVER KONVER KONVER
P		M	First entry of 3 word in-plane control vector w		/MATS	/(1)	ALGCON ALGCON	M M	P P1
PA1		O	Not used		/MATS	/(259)	ALGCON	O	PA1
PA2		D	$-K^{(2)}_w/K^{(2)}_{\delta_E}$		/MATS	/(260)	ALGCON AL1	O M	PA2 PA2
PDA	$K^{(3)}_{\lambda_\psi}$	I	Explicit partial of $K^{(3)}$ wrt λ_ψ when w is optimal.		/MATS	/(18)	ALGCON AL1	I O	PDA PDA
PROD1	$\partial(\delta p/\delta y)/\partial y$	I	A 2x64 matrix that contains the total first partials of the matrix DEPDEY wrt the QL state.		/MATS	/(310)	ALGCON ALGCON APPLY APPLY	I I I I	PROD1 PROD5 PROD1 PROD5
PROD5	$\partial(\delta p/\delta y)/\partial y$	I	A 2x64 matrix that contains the total first partials of the matrix DEPDEY wrt the QL state.		/MATS	/(310)	ALGCON ALGCON APPLY APPLY	I I I I	PROD1 PROD5 PROD1 PROD5
PROD9		I	A 2x64 matrix that contains $\partial(\delta p/\delta y)/\partial\lambda_v$, $\partial(\delta p/\delta y)/\partial\lambda_\gamma$ and $\partial(\delta p/\delta y)/\partial\lambda_\psi$.		/MATS	/(502)	ALGCON APPLY	I I	PROD9 PROD9
P1		M	First entry of 3 word in-plane control vector w		/MATS	/(1)	ALGCON ALGCON	M M	P P1
SKWW		W	A 3x9 array that contains Kww		/ALGCON/(*	)	ALGCON	W	SKWW
SKYYI		W	A 3x8 array that contains Kyy_i		/ALGCON/(*	)	ALGCON	W	SKYYI
SLWV		W	A 2x6 array that contains Lwp, where $L = (K^{(1)}, K^{(2)})^T$		/ALGCON/(*	)	ALGCON	W	SLWV
SLYLI		I	A 2x8 array that contains $Ly\lambda_i$		/ALGCON/(*	)	ALGCON	I	SLYLI
SLYYI		W	A 2x8 array that contains Lyy_i		/ALGCON/(*	)	ALGCON	W	SLYYI

867

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T	T	M	Thrust (LBS)	/DYNA /(	42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
VDA	$K^{(3)}_{\lambda_v}$	I	Explicit partial of $K^{(3)}$ wrt λ_v when ω is optimal.	/MATS /(	16)	ALGCON	I	VDA
						AL1	O	VDA
XKL11		C	The first word in a 3x3 array that contains K_{λ_v}, K_{λ_z} and K_{λ_ψ}	/ALGCON/(*	)	ALGCON	C	XKL11
XKPI11		M	The first entry in the 3x3 matrix $-K^{-1}_{\omega}$	/MATS /(	250)	ALGCON	M	XKPI11
XKWW	$K^{(1)}_{TT}$	I	First entry in 3x6 matrix containing $K_{\omega\omega}$	/MATS /(	23)	ALGCON	I	XKWW
						TH3	O	XK1TT
XKWY	$K^{(1)}_{VT}$	I	The first entry in a 3x3 matrix containing $K_{v\omega}$	/MATS /(	70)	ALGCON	I	XKWY
						TH3	M	XK1VT
XKY	$K^{(1)}_{V}$	I	The first entry in a 3x8 matrix containing K_y	/MATS /(	46)	ALGCON	I	XKY
						ALGCON	I	XK1V
						TH3	M	XK1V
						TH4	O	XK1V
XKYY	$K^{(1)}_{vv}$	I	The first entry in a 3x8 matrix containing K_{yv}	/MATS /(	142)	ALGCON	I	XKYY
						TH3	M	XK1VV
						TH4	O	XK1VV
XK1		I	First entry of 3 word in-plane control constraints K	/MATS /(	4)	ALGCON	I	XK1
						THROTL	I	XK1
						TH1	O	XK1
						TH2	O	XK1
						TH3	O	XK1
						TH4	O	XK1
XK1T	$K^{(1)}_{T}$	I	First entry of 3x3 matrix containing the explicit partials of K with respect to ω, K_{ω}	/MATS /(	7)	ALGCON	I	XK1T
						TH1	O	XK1T
						TH2	O	XK1T
						TH3	M	XK1T
XK1V	$K^{(1)}_{V}$	I	The first entry in a 3x8 matrix containing K_y	/MATS /(	46)	ALGCON	I	XKY
						ALGCON	I	XK1V
						TH3	M	XK1V
						TH4	O	XK1V
VKPI		I	A 2x2 array that contains $-[L_p]^{-1}$	/ALGCON/(*	)	ALGCON	I	VKPI

SUBRØUTINE
AL1

Purpose

AL1 evaluates the optimal angle of attack constraint, Equation 16.8-3 in Vol. I. In addition, it computes the explicit first partials of this constraint with respect to the state, costate and in-plane control as they are needed.

AL1

```
      SUBROUTINE AL1                                                    AL1
C                                                                       AL1
C                    THIS ROUTINE APPLYS WHEN THE ANGLE OF ATTACK IS    AL1
C                    OPTIMAL.  THIS ENTRY COMPUTES THE EXPICIT PARTIALS AL1
C                    OF THE CONSTRAINING EQUATION WITH RESPECT TO THE   AL1
C                    STATE, COSTATE AND IN-PLANE CONTROL.               AL1
C                                                                       AL1
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,A, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK10  ,XK20  ,XK30  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
```

```
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,      MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,      MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,      MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,      MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,      MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,      MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                 MATS
      COMMON /MATS/                                                        MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)     MATS
      COMMON /MATS/                                                        MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,      MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,      MATS
     *POR   ,POO   ,PLG   ,PLP                                             MATS
      EQUIVALENCE(PROD1,PROD5)                                             MATS
      ENTRY AL1010                                                         AL1
      ASSIGN 1001 TO LABL1                                                 AL1
      GO TO 1000                                                           AL1   1000
C                    THIS ENTRY COMPUTES THE FIRST PARTIALS WITH RESPECT   AL1
C                    TO IN-PLANE CONTROL.                                  AL1
      ENTRY AL1001                                                         AL1
      ASSIGN 1002 TO LABL1                                                 AL1
      GO TO 1000                                                           AL1   1000
C                    THIS ENTRY EVALUATES THE CONSTRAINING EQ. ONLY.       AL1
      ENTRY AL1000                                                         AL1
      ASSIGN 1003 TO LABL1                                                 AL1
C                    INITIALIZATION FOR ALL ENTRIES.                       AL1
 1000 VLV = V*LV                                                           AL1
      XLTOT = LGAM*COSPHI + LPSI*SINPHI/COSGAM                             AL1
      XDEL = T*SIDAE                                                       AL1
      YDEL = -T*CODAE                                                      AL1
      PA2 = -XK2A/XK2D                                                     AL1
      OMPA2 = 1. - PA2                                                     AL1
      DXDA = -XDEL*OMPA2 + DB*SINA - DRAGA                                 AL1
      DYDA = -YDEL*OMPA2 - DB*COSA + LIFTA                                 AL1
      GO TO LABL1                                                          AL1
C                    COMPUTE PARTIALS WITH RESPECT TO STATE/COSTATE        AL1
 1001 DPA2  = (XK2VA + XK2VD*PA2)/XK2D                                     AL1
      DDX = -XDEL*DPA2 - DRAGVA                                            AL1
      DDY = -YDEL*DPA2 + LIFTVA                                            AL1
      XK3V = LV*(DXDA + V*DDX) + XLTOT*DDY                                 AL1
      XK3G = DYDA*LPSI*SINPHI*SINGAM/COSGAM**2                             AL1
      DPA2 = (XK2RA + XK2RD*PA2)/XK2D                                      AL1
      DDX = -XDEL*DPA2 + DBR*SINA - DRAGRA                                 AL1
      DDY = -YDEL*DPA2 - DBR*COSA + LIFTRA                                 AL1
      XK3R = VLV*DDX + XLTOT*DDY                                           AL1
      DPA2 = (XK2MA + XK2MD*PA2)/XK2D                                      AL1
      DDX = -XDEL*DPA2                                                     AL1
      DDY = -YDEL*DPA2 + LIFTMA                                            AL1
      XK3M = VLV*DDX + XLTOT*DDY                                           AL1
      VDA = V*DXDA                                                         AL1
      GDA = COSPHI*DYDA                                                    AL1
      PDA = SINPHI/COSGAM*DYDA                                             AL1
C                    COMPUTE PARTIALS WITH RESPECT TO IN-PLANE CONTROL.    AL1
 1002 DPA2 = (XK2TA + XK2TD*PA2)/XK2D                                      AL1
      DDX = -SIDAE*OMPA2 - XDEL*DPA2                                       AL1
      DDY =  CODAE*OMPA2 - YDEL*DPA2                                       AL1
      XK3T = VLV*DDX + XLTOT*DDY                                           AL1
      DPA2 = (XK2DA + XK2DD*PA2)/XK2D                                      AL1
      DDX = -YDEL*OMPA2 - XDEL*DPA2                                        AL1
      DDY =  XDEL*OMPA2 - YDEL*DPA2                                        AL1
      XK3D = VLV*DDX + XLTOT*DDY                                           AL1
      DPA2 = (XK2AA + XK2DA*PA2)/XK2D                                      AL1
      DDX =  YDEL*OMPA2 - XDEL*DPA2 + DB*COSA - DRAGAA                     AL1
      DDY = -XDEL*OMPA2 - YDEL*DPA2 + DB*SINA + LIFTAA                     AL1
      XK3A = VLV*DDX + XLTOT*DDY                                           AL1
C                    EVALUATE CONSTRAINING EQUATION.                       AL1
 1003 XK3 = VLV*DXDA + XLTOT*DYDA                                          AL1
      RETURN                                                               AL1
      END                                                                  AL1
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA	/(151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA	/(10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol		/DYNA	/(4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCOL	I	COSGAM
							STATEF	M	COSGAM
COSPHI	$\cos\phi$	I	See symbol		/DYNA	/(93)	AL1	I	COSPHI
							AL4	I	COSPHI
							APPLY	I	COSPHI
							ARCIN	O	COSPHI
							CONTRL	M	COSPHI
							OUTPUT	I	COSPHI
DB	D_b	I	Base drag	(LBS)	/DYNA	/(163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DRAGA	$\partial D/\partial\alpha$	I	See symbol		/DYNA	/(72)	AL1	I	DRAGA
							AL5	I	DRAGA
							AL7	I	DRAGA
							AL8	I	DRAGA
							AL9	I	DRAGA
							APPLY	I	DRAGA
							TH3	I	DRAGA
							UT	M	DRAGA

872

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DRAGAA	$\partial^2 D/\partial\alpha^2$	I	See symbol	/DYNA	/(78)	AL1	I	DRAGAA
						AL5	I	DRAGAA
						AL7	I	DRAGAA
						AL8	I	DRAGAA
						AL9	I	DRAGAA
						APPLY	I	DRAGAA
						TH3	I	DRAGAA
						UT	M	DRAGAA
DRAGRA	$\partial^2 D/\partial R\partial\alpha$	I	See symbol	/DYNA	/(77)	AL1	I	DRAGRA
						AL5	I	DRAGRA
						AL7	I	DRAGRA
						AL8	I	DRAGRA
						AL9	I	DRAGRA
						APPLY	I	DRAGRA
						TH3	I	DRAGRA
						UT	M	DRAGRA
DRAGVA	$\partial^2 D/\partial V\partial\alpha$	I	See symbol	/DYNA	/(75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
GDA	$\kappa^{(3)}_{\lambda_\gamma}$	O	Explicit partial of $\kappa^{(3)}$ wrt λ_γ when α is optimal.	/MATS	/(17)	ALGCON	I	GDA
						AL1	O	GDA
LGAM	λ_γ	I	Relative flight path angle costate	/D	/(101)	AL1	I	LGAM
						ARCIN	I	LGAM
						CONTRL	I	LGAM
						NLDRV	I	LGAM
						OUTPUT	I	LGAM
						WRAPUP	I	LGAM
LIFTA	$\partial L/\partial\alpha$	I	See symbol	/DYNA	/(63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTAA	$\partial^2 L/\partial\alpha^2$	I	See symbol	/DYNA	/(144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA
LIFTMA	$\partial^2 L/\partial m\partial\alpha$	I	See symbol	/DYNA	/(85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA
LIFTRA	$\partial^2 L/\partial R\partial\alpha$	I	See symbol	/DYNA	/(68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTVA	$\partial^2 L/\partial V\partial\alpha$	I	See symbol	/DYNA	/(66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA

874

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LPSI	λ_ψ	I	Relative azimuth angle costate	/D	/(102)	AL1	I	LPSI
						ARCIN	I	LPSI
						CONTRL	I	LPSI
						NLDRV	I	LPSI
						OUTPUT	I	LPSI
						WRAPUP	I	LPSI
LV	λ_v	I	Relative velocity costate	/D	/(100)	AL1	I	LV
						CONTRL	I	LV
						NLDRV	I	LV
						OUTPUT	I	LV
						WRAPUP	I	LV
PA2		M	$-K^{(2)}_{\alpha}/K^{(2)}_{\delta_E}$	/MATS	/(260)	ALGCON	O	PA2
						AL1	M	PA2
PDA	$K^{(3)}_{\lambda_\psi}$	O	Explicit partial of $K^{(3)}$ wrt λ_ψ when α is optimal.	/MATS	/(18)	ALGCON	I	PDA
						AL1	O	PDA
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA	/(152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA	/(9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol	/DYNA	/(3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM
SINPHI	$\sin\phi$	I	See symbol	/DYNA	/(92)	AL1	I	SINPHI
						AL4	I	SINPHI
						APPLY	I	SINPHI
						CONTRL	M	SINPHI
						OUTPUT	I	SINPHI
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
V	V	I	Relative velocity.	(FT/SEC)	/D	/(91)	AL1	I	V
							AL4	I	V
							AL7	I	V
							AL8	I	V
							AL9	I	V
							BCOND	I	NOM
							BNDRV	O	NOM
							BRANPT	M	NOM
							CONTRL	I	V
							ENDPT	I	NOM
							ENVPRQ	I	V
							FETCH	O	NOM
							INTERP	M	V
							INTRPT	M	NOM
							NLDRV	O	NOM
							NLDRV	I	V
							OUTPUT	I	V
							PDBCOL	I	V
							STATEF	I	V
							WRAPUP	I	V
VDA	$K^{(3)}_{\lambda_v}$	O	Explicit partial of $K^{(3)}$ wrt λ_v when α is optimal.		/MATS	/(16)	ALGCON	I	VDA
							AL1	O	VDA

SUBRØUTINE AL2

Purpose

AL2 evaluates the constant angle of attack constraint, Equation 5.2-1 in Vol. I.

```
      SUBROUTINE AL2                                                    AL2
C                                                                       AL2
C                 THIS ROUTINE APPLIES WHEN ALPHA IS A CONSTANT         AL2
C                                                                       AL2
C                                                                       AL2
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
```

```
     *PV     ,PG     ,PP     ,PR     ,PO     ,PVV    ,PGV    ,PPV    ,PRV    ,      MATS
     *POV    ,PGG    ,PPG    ,PRG    ,POG    ,PPP    ,PRP    ,POP    ,PRR    ,      MATS
     *POR    ,POO    ,PLG    ,PLP                                                   MATS
      EQUIVALENCE(PROD1,PROD5)                                                      MATS
C                                                                                   AL2
      ENTRY AL2002                                                                  AL2
      ENTRY AL2001                                                                  AL2
   40 XK3A = 1.                                                                     AL2
      ENTRY AL2000                                                                  AL2
   50 XK3 = ALPHA - CALPHA                                                          AL2
C                                                                                   AL2
      RETURN                                                                        AL2
      END                                                                           AL2
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	I	Angle of attack (RAD)	/DYNA	/(79)	AEROCO	I	ALPHA
						ALGCON	M	ALPHA
						AL2	I	ALPHA
						ARCIN	M	ALPHA
						CONTRL	M	ALPHA
						ENVPRO	I	ALPHA
						MOMECO	I	ALPHA
						NPLANE	I	ALPHA
						OUTPUT	I	ALPHA
						TRAJIN	O	ALPHA
						UT	I	ALPHA
						WRAPUP	I	ALPHA
CALPHA	C_α	I	Value for angle of attack in case constant angle of attack constraint is used. (RADS)	/DYNA	/(161)	AL2	I	CALPHA
						NPLANE	O	CALPHA

128

SUBRØUTINE
AL3

Purpose

AL3 evaluates the untrimmed lift constraint, Equation 10.4-1 in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL3                                                    AL3
C                                                                       AL3
C                    THIS ROUTINE APPLIES WHEN THE UNTRIMMED LIFT LIMIT AL3
C                    IS IN EFFFCT.                                      AL3
C                                                                       AL3
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
```

```
          *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,      MATS
          *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,      MATS
          *POR   ,POO   ,PLG   ,PLP                                            MATS
           EQUIVALENCE(PROD1,PROD5)                                            MATS
C                                                                              AL3
C                        THIS ENTRY COMPUTES EXPLICIT SECOND PARTIALS OF THE   AL3
C                        CONSTRAINING EQ. W/RESPECT TO STATE                   AL3
           ENTRY AL3020                                                        AL3
           XK3VV = ULFTVV                                                      AL3
           XK3RV = ULFTVR                                                      AL3
           XK3RR = ULFTRR                                                      AL3
C                        THIS ENTRY COMPUTES MIXED SECOND PARTS. W/RESPECT     AL3
C                        TO STATE AND CONTROL.                                 AL3
           ENTRY AL3011                                                        AL3
           XK3VA = ULFTVA                                                      AL3
           XK3RA = ULFTRA                                                      AL3
C                        THIS ENTRY COMPUTES FIRST PARTS W/RESPECT TO STATE    AL3
           ENTRY AL3010                                                        AL3
           XK3V = ULFTV                                                        AL3
           XK3R = ULFTR                                                        AL3
C                        THIS ENTRY COMP. 2ND PARTS W/RESP. TO CONTROL.        AL3
           ENTRY AL3002                                                        AL3
        30 XK3AA = ULFTAA                                                      AL3
C                        THIS ENTRY COMP. 1ST PARTS W/RESP. TO CONTROL         AL3
           ENTRY AL3001                                                        AL3
        40 XK3A = ULFTA                                                        AL3
C                        THIS ENTRY EVAL. COSTRAINING EQ. ONLY                 AL3
           ENTRY AL3000                                                        AL3
        50 XK3 = ULFT - CULFT                                                  AL3
C                                                                              AL3
           RETURN                                                              AL3
           END                                                                 AL3
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK		LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CULFT	C_{L_u}	I	Magnitude of untrimmed lift limit	(LBS)	/DYNA	/(	165)	AL3 NPLANE	I O	CULFT CULFT
ULFT	L_u	I	Untrimmed aerodynamic lift	(LBS)	/DYNA	/(	164)	AL3 NPLANE UT	I I M	ULFT ULFT ULFT
ULFTA	$\partial L_u/\partial\alpha$	I	See symbol		/DYNA	/(	166)	AL3 UT	I M	ULFTA ULFTA
ULFTAA	$\partial^2 L_u/\partial\alpha^2$	I	See symbol		/DYNA	/(	178)	AL3 UT	I M	ULFTAA ULFTAA
ULFTR	$\partial L_u/\partial R$	I	See symbol		/DYNA	/(	133)	AL3 UT	I M	ULFTR ULFTR
ULFTRA	$\partial^2 L_u/\partial R\partial\alpha$	I	See symbol		/DYNA	/(	138)	AL3 UT	I M	ULFTRA ULFTRA
ULFTRR	$\partial^2 L_u/\partial R^2$	I	See symbol		/DYNA	/(	137)	AL3 UT	I M	ULFTRR ULFTRR
ULFTV	$\partial L_u/\partial V$	I	See symbol		/DYNA	/(	132)	AL3 UT	I M	ULFTV ULFTV
ULFTVA	$\partial^2 L_u/\partial V\partial\alpha$	I	See symbol		/DYNA	/(	136)	AL3 UT	I M	ULFTVA ULFTVA
ULFTVR	$\partial^2 L_u/\partial V\partial R$	I	See symbol		/DYNA	/(	135)	AL3 UT	I M	ULFTVR ULFTVR
ULFTVV	$\partial^2 L_u/\partial V^2$	I	See symbol		/DYNA	/(	134)	AL3 UT	I M	ULFTVV ULFTVV

SUBRØUTINE AL4

Purpose

AL4 evaluates the vertical rise and pitchover constraint, Equation 10.1-1 in Vol. I. Moreover, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL4                                                    AL4
C                                                                       AL4
C                    THIS ROUTINE APPLIES FOR VERT. RISE AND PICHOVER   AL4
C                                                                       AL4
C                                                                       AL4
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX     ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,  JUL21
     *COSA   ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,  DYNA
     *ROR    ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,  DYNA
     *QV     ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,  DYNA
     *FVACT  ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,  DYNA
     *ISPV   ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,  DYNA
     *ISPRM  ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,  DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W      ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,  DYNA
     *MUR    ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,  DYNA
     *XCGMM  ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,  DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM   ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,  DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM  ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,  DYNA
     *SID    ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,  DYNA
     *DB     ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED   DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT    ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,  DYNA
     *ILOAD  ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,  DYNA
     *CLMM   ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,  DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM   ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,  JUL21
     *SFCV   ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1     ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,  MATS
     *XK1D   ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,  MATS
     *XM19   ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,  MATS
     *XK3TD  ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,  MATS
     *XK3DA  ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,  MATS
     *XK1V   ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,  MATS
     *XK1R   ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,  MATS
     *XK1M   ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,  MATS
     *XK1VD  ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,  MATS
     *XK1GD  ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,  MATS
     *XK1PD  ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,  MATS
     *XK1RD  ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,  MATS
     *XK1OD  ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,  MATS
     *XK1UD  ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT    MATS
      COMMON /MATS/                                                     MATS
     *XK1MD  ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,  MATS
     *XK1ZD  ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,  MATS
     *XK1GV  ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,  MATS
     *XK1OV  ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,  MATS
     *XK1ZV  ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,  MATS
     *XK1RG  ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,  MATS
     *XK1MG  ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,  MATS
     *XK1RP  ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,  MATS
     *XK1MP  ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,  MATS
     *XK1OR  ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,  MATS
```

```
     *XK1ZR ,XK2ZR ,XK3ZR ,XK100 ,XK200 ,XK300 ,XK1U0 ,XK2U0 ,XK3U0 ,    MATS
     *XK1M0 ,XK2M0 ,XK3M0 ,XK1Z0 ,XK2Z0 ,XK3Z0 ,XK1UU ,XK2UU ,XK3UU ,    MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,    MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,    MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPOY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,    MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,    MATS
     *POR   ,POO   ,PLG   ,PLP                                          MATS
      EQUIVALENCE(PROD1,PROD5)                                          MATS
C                                                                       AL4
      ASF(X, Y) = (XKG*Y - XKP*X)/SQUARE                                AL4
      BSF(E, F, G, H) = (Y2MX2*(E*H + G*F) + TWOXY*(E*G - F*H))/SQSQ    AL4
C                     THIS ENTRY COMPUTES 2ND PARTS. W/RESP. TO STATE   AL4
      ENTRY AL4020                                                      AL4
      ASSIGN 6 TO IGO                                                   AL4
      ASSIGN 3 TO LABL                                                  AL4
      GO TO 2                                                           AL4
C                     THIS ENTRY COMPUTES MIXED 2ND PARTS. W/RESP. TO   AL4
C                     STATE AND CONTROL                                 AL4
      ENTRY AL4011                                                      AL4
      ASSIGN 10 TO IGO                                                  AL4
      ASSIGN 5 TO LABL                                                  AL4
      GO TO 4                                                           AL4
C                     THIS ENTRY COMP. 1ST PARTS. W/RESP. TO STATE      AL4
      ENTRY AL4010                                                      AL4
      ASSIGN 20 TO IGO                                                  AL4
      ASSIGN 5 TO LABL                                                  AL4
      GO TO 4                                                           AL4
C                     THIS ENTRY COMP. 2ND PARTS. W/RESP. TO CONTROL    AL4
      ENTRY AL4002                                                      AL4
      ASSIGN 30 TO IGO                                                  AL4
      GO TO 5                                                           AL4
C                     THIS ENTRY COMP. 1ST PARTS. W/RESP. TO CONTROL    AL4
      ENTRY AL4001                                                      AL4
      ASSIGN 40 TO IGO                                                  AL4
      GO TO 5                                                           AL4
C                     THIS ENTRY EVAL. COSTRAINING EQ. ONLY             AL4
      ENTRY AL4000                                                      AL4
      ASSIGN 50 TO IGO                                                  AL4
      GO TO 5                                                           AL4
C                     INITIALIZATION FOR 2ND PARTS W/RESP TO STATE      AL4
    2 Y2MX2 = (XKP - XKG)*(XKP + XKG)                                   AL4
      TWOXY = 2.*XKG*XKP                                                AL4
      YYRR = LIFTRR - DBRR*SINA                                         AL4
C                     COMP. EXPLICIT 2ND PARTS W/RESP. TO STATE OF KINE- AL4
C                     MATICS                                            AL4
      XKGVV = 2./R*COSGAM                                               AL4
      XKPVV = 2./(R*COSRHO)*COSGAM**2*SINRHO*SINPSI                     AL4
      XKGVG = -2./R*V*SINGAM                                            AL4
      XKPVG =-2.*(SINRHO*SINGAM*(2./(R*COSRHO)*V*COSGAM*SINPSI + OMEGA) AL4
     *           + OMEGA*COSRHO*COSPSI*COSGAM)                          AL4
      XKGVP = 2.*OMEGA*COSRHO*COSPSI                                    AL4
      XKPVP = 2.*(V/(R*COSRHO)*COSGAM**2*SINRHO*COSPSI + OMEGA*COSRHO   AL4
     *           *SINPSI*SINGAM)                                        AL4
      XKGVR = -2./R**2*V*COSGAM                                         AL4
      XKPVR = -2./(R**2*COSRHO)*V*COSGAM**2*SINRHO*SINPSI               AL4
      XKGVO = -2.*OMEGA*SINRHO*SINPSI                                   AL4
      XKPVO = 2.*(V/(R*COSRHO**2)*COSGAM**2*SINPSI + OMEGA*(COSRHO      AL4
     *           *COSGAM + SINRHO*COSPSI*SINGAM))                       AL4
      XKGGG = (G - V**2/R)*COSGAM - R*OMEGA2*COSRHO*(COSRHO*COSGAM      AL4
     *           + SINRHO*COSPSI*SINGAM)                                AL4
      XKPGG = -2.*V*(V/(R*COSRHO)*COS2GM*SINRHO*SINPSI + OMEGA*(SINRHO  AL4
     *           *COSGAM - COSRHO*COSPSI*SINGAM))                       AL4
      XKGGP = -R*OMEGA2*COSRHO*SINRHO*SINPSI*COSGAM                     AL4
      XKPGP = -2.*V*(V/(R*COSRHO)*SINGAM*COSGAM*COSPSI*SINRHO - OMEGA   AL4
     *           *COSRHO*SINPSI*COSGAM)                                 AL4
      XKGGR = (V**2  - 2.*G*R)/R**2*SINGAM - OMEGA2*COSRHO*(COSRHO      AL4
     *           *SINGAM - SINRHO*COSPSI*COSGAM)                        AL4
      XKPGR = 2./COSRHO*(V/R)**2*SINRHO*SINGAM*COSGAM*SINPSI            AL4
      XKGGO = R*OMEGA2*(SIN2RO*SINGAM + COS2RO*COSPSI*COSGAM)           AL4
      XKPGO = -2.*V*(V/(R*COSRHO**2)*SINGAM*COSGAM*SINPSI + OMEGA       AL4
     *           *(COSRHO*SINGAM - SINRHO*COSPSI*COSGAM))               AL4
```

```
      XKGPP = -OMEGA*COSRHO*(2.*V*SINPSI + R*OMEGA*SINRHO*COSPSI*SINGAM) AL4
      XKPPP = -(V*COSGAM)**2/(R*COSRHO)*SINRHO*SINPSI + OMEGA*COSRHO    AL4
     *        *(2.*V*COSPSI*SINGAM - R*OMEGA*SINRHO*SINPSI)             AL4
      XKGPR = -OMEGA2*COSRHO*SINRHO*SINPSI*SINGAM                      AL4
      XKPPR = SINRHO*COSPSI*(OMEGA2*COSRHO - (V/R*COSGAM)**2/COSRHO)   AL4
      XKGPO = -OMEGA*(R*OMEGA*COS2RO*SINPSI*SINGAM + 2.*V*SINRHO*COSPSI) AL4
      XKPPO = COSPSI*((V/COSRHO*COSGAM)**2/R + R*OMEGA2*COS2RO) - 2.*V  AL4
     *        *OMEGA*SINRHO*SINPSI*SINGAM                               AL4
      XKGRR = 2./R**3*(V**2 - 3.*G*R)*COSGAM                           AL4
      XKPRR = 2.*SINRHO*SINPSI*(V/R*COSGAM)**2/(R*COSRHO)              AL4
      XKGRO = OMEGA2*(COS2RO*COSPSI*SINGAM - SIN2RO*COSGAM)            AL4
      XKPRO = SINPSI*(OMEGA2*COS2RO - (V/R*COSGAM/COSRHO)**2)          AL4
      XKGOO = -2.*OMEGA*(R*OMEGA*(COS2RO*COSGAM + SIN2RO*COSPSI*SINGAM) AL4
     *        + V*COSRHO*SINPSI)                                        AL4
      XKPOO = 2.*(V*COSGAM*SINRHO*(V/(R*COSRHO**3)*COSGAM*SINPSI       AL4
     *        - OMEGA) + OMEGA*(V*COSRHO*COSPSI*SINGAM - R*OMEGA*SINPSI AL4
     *        *SIN2RO))                                                 AL4
      GO TO 4                                                           AL4
C                   COMPUTE EXPLICIT 2ND PARTS W/RESP. TO STATE OF BANK AL4
C                   ANGLE                                               AL4
    3 PVV = ASF(XKGVV, XKPVV) + BSF(XKGV, XKPV, XKGV, XKPV)             AL4
      PGV = ASF(XKGVG, XKPVG) + BSF(XKGV, XKPV, XKGG, XKPG)             AL4
      PPV = ASF(XKGVP, XKPVP) + BSF(XKGV, XKPV, XKGP, XKPP)             AL4
      PRV = ASF(XKGVR, XKPVR) + BSF(XKGV, XKPV, XKGR, XKPR)             AL4
      POV = ASF(XKGVO, XKPVO) + BSF(XKGV, XKPV, XKGO, XKPO)             AL4
      PGG = ASF(XKGGG, XKPGG) + BSF(XKGG, XKPG, XKGG, XKPG)             AL4
      PPG = ASF(XKGGP, XKPGP) + BSF(XKGG, XKPG, XKGP, XKPP)             AL4
      PRG = ASF(XKGGR, XKPGR) + BSF(XKGG, XKPG, XKGR, XKPR)             AL4
      POG = ASF(XKGGO, XKPGO) + BSF(XKGG, XKPG, XKGO, XKPO)             AL4
      PPP = ASF(XKGPP, XKPPP) + BSF(XKGP, XKPP, XKGP, XKPP)             AL4
      PRP = ASF(XKGPR, XKPPR) + BSF(XKGP, XKPP, XKGR, XKPR)             AL4
      POP = ASF(XKGPO, XKPPO) + BSF(XKGP, XKPP, XKGO, XKPO)             AL4
      PRR = ASF(XKGRR, XKPRR) + BSF(XKGR, XKPR, XKGR, XKPR)             AL4
      POR = ASF(XKGRO, XKPRO) + BSF(XKGR, XKPR, XKGO, XKPO)             AL4
      POO = ASF(XKGOO, XKPOO) + BSF(XKGO, XKPO, XKGO, XKPO)             AL4
C                   COMP. EXPLICIT 2ND PARTS W/RESP TO STATE OF COSINE  AL4
C                   OF BANK ANGLE                                       AL4
      CPVV = -COSPHI*PV*PV - SINPHI*PVV                                 AL4
      CPVG = -COSPHI*PV*PG - SINPHI*PGV                                 AL4
      CPVP = -COSPHI*PV*PP - SINPHI*PPV                                 AL4
      CPVR = -COSPHI*PV*PR - SINPHI*PRV                                 AL4
      CPVO = -COSPHI*PV*PO - SINPHI*POV                                 AL4
      CPGG = -COSPHI*PG*PG - SINPHI*PGG                                 AL4
      CPGP = -COSPHI*PG*PP - SINPHI*PPG                                 AL4
      CPGR = -COSPHI*PG*PR - SINPHI*PRG                                 AL4
      CPGO = -COSPHI*PG*PO - SINPHI*POG                                 AL4
      CPPP = -COSPHI*PP*PP - SINPHI*PPP                                 AL4
      CPPR = -COSPHI*PP*PR - SINPHI*PRP                                 AL4
      CPPO = -COSPHI*PP*PO - SINPHI*POP                                 AL4
      CPRR = -COSPHI*PR*PR - SINPHI*PRR                                 AL4
      CPRO = -COSPHI*PR*PO - SINPHI*POR                                 AL4
      CPOO = -COSPHI*PO*PO - SINPHI*POO                                 AL4
      GO TO 5                                                           AL4
C                   INITIALIZATION FOR 1ST PARTS W/RESP TO STATE AND/OR AL4
C                   MIXED 2ND PARTS.                                    AL4
    4 SQUARE = XKG**2 + XKP**2                                          AL4
      SQSQ = SQUARE**2                                                  AL4
      YYR = LIFTR - DBR*SINA                                            AL4
C                   COMPUTE EXPLICIT 1ST PARTS W/RESP TO STATE OF KINE- AL4
C                   MATICS                                              AL4
      XKGV = 2.*(V/R*COSGAM + OMEGA*COSRHO*SINPSI) - GAMMAD             AL4
      XKGG = (G - V**2/R)*SINGAM - R*OMEGA2*COSRHO*(COSRHO*SINGAM       AL4
     *        - SINRHO*COSPSI*COSGAM)                                   AL4
      XKGP = OMEGA*COSRHO*(2.*V*COSPSI - R*OMEGA*SINRHO*SINPSI*SINGAM)  AL4
      XKGR = (2.*G *R- V**2)/R**2*COSGAM + OMEGA2*COSRHO*(COSRHO*COSGAM AL4
     *        + SINRHO*COSPSI*SINGAM)                                   AL4
      XKGO = -OMEGA*(R*OMEGA*(SIN2RO*COSGAM - COS2RO*COSPSI*SINGAM) + 2. AL4
     *        *V*SINRHO*SINPSI)                                         AL4
      XKPV = 2.*(COSGAM*SINRHO*(V/(R*COSRHO)*COSGAM*SINPSI + OMEGA)     AL4
     *        - OMEGA*COSRHO*COSPSI*SINGAM)                             AL4
      XKPG = -2.*V*(SINRHO*SINGAM*(V/(R*COSRHO)*COSGAM*SINPSI + OMEGA)  AL4
     *        + OMEGA*COSRHO*COSPSI*COSGAM)                             AL4
      XKPP = (V*COSGAM)**2/(R*COSRHO)*SINRHO*COSPSI + OMEGA*COSRHO      AL4
     *        *(R*OMEGA*SINRHO*COSPSI + 2.*V*SINPSI*SINGAM)             AL4
      XKPR = SINRHO*SINPSI*(OMEGA2*COSRHO - (V/R*COSGAM)**2/COSRHO)     AL4
```

```
      XKPO = V*COSGAM*(V/(R*COSRHO**2)*COSGAM*SINPSI + 2.*OMEGA*COSRHO)  AL4
     *         + OMEGA*(R*OMEGA*SINPSI*COS2RO + 2.*V*SINRHO*COSPSI*SINGAM) AL4
C                    COMPUTE 1ST PARTS W/RESP TO STATE OF BANK ANGLE      AL4
      PV = ASF(XKGV, XKPV)                                                AL4
      PG = ASF(XKGG, XKPG)                                                AL4
      PP = ASF(XKGP, XKPP)                                                AL4
      PR = ASF(XKGR, XKPR)                                                AL4
      PO = ASF(XKGO, XKPO)                                                AL4
C                    COMPUTE 1ST PARTS W/RESP TO STATE OF COSINE OF BANK  AL4
      CPV = -SINPHI*PV                                                    AL4
      CPG = -SINPHI*PG                                                    AL4
      CPP = -SINPHI*PP                                                    AL4
      CPR = -SINPHI*PR                                                    AL4
      CPO = -SINPHI*PO                                                    AL4
      GO TO LABL                                                          AL4
C                    INITIALIZATION FOR ALL ENTRIES                       AL4
    5 TCDAE = T*CODAE                                                     AL4
      TSDAE = T*SIDAE                                                     AL4
      DBCA = DB*COSA                                                      AL4
      DBSA = DB*SINA                                                      AL4
      YY = TSDAE + LIFT - DBSA                                            AL4
      ZZ = TCDAE + LIFTA - DBCA                                           AL4
      GO TO IGO                                                           AL4
C                    2ND PARTS W/RESP TO STATE                            AL4
    6 XK3VV = LIFTVV*COSPHI + 2.*LIFTV*CPV + YY*CPVV + M*XKGVV            AL4
      XK3GV = LIFTV*CPG + YY*CPVG + M*XKGVG                               AL4
      XK3PV = LIFTV*CPP + YY*CPVP + M*XKGVP                               AL4
      XK3RV = LIFTV*CPR + LIFTVR*COSPHI +YYR*CPV + YY*CPVR + M*XKGVR      AL4
      XK3OV = LIFTV*CPO + YY *CPVO + M*XKGVO                              AL4
      XK3MV = XKGV + LIFTM*CPV + LIFTVM*COSPHI                            AL4
      XK3GG = YY*CPGG + M*XKGGG                                           AL4
      XK3PG = YY*CPGP + M*XKGGP                                           AL4
      XK3RG = YY*CPGR + M*XKGGR + YYR*CPG                                 AL4
      XK3OG = YY*CPGO + M*XKGGO                                           AL4
      XK3MG = XKGG + LIFTM*CPG                                            AL4
      XK3PP = YY*CPPP + M*XKGPP                                           AL4
      XK3RP = YY*CPRR + M*XKGPR + YYR*CPP                                 AL4
      XK3OP = YY*CPPO + M*XKGPO                                           AL4
      XK3MP = XKGP + LIFTM*CPP                                            AL4
      XK3RR = YYRR*COSPHI + 2.*YYR*CPR + YY*CPRR + M*XKGRR                AL4
      XK3OR = YYR*CPO + YY*CPRO + M*XKGRO                                 AL4
      XK3MR = XKGR + LIFTM*CPR + LIFTRM*COSPHI                            AL4
      XK3OO = YY*CPOO + M*XKGOO                                           AL4
      XK3MO = XKGO + LIFTM*CPO                                            AL4
      XK3MM = LIFTMM*COSPHI                                               AL4
C                    MIXED PARTS W/RESP TO STATE AND CONTROL              AL4
   10 XK3VT = SIDAE*CPV                                                   AL4
      XK3GT = SIDAE*CPG                                                   AL4
      XK3PT = SIDAE*CPP                                                   AL4
      XK3RT = SIDAE*CPR                                                   AL4
      XK3OT = SIDAE*CPO                                                   AL4
      XK3VD = -TCDAE*CPV                                                  AL4
      XK3GD = -TCDAE*CPG                                                  AL4
      XK3PD = -TCDAE*CPP                                                  AL4
      XK3RD = -TCDAE*CPR                                                  AL4
      XK3OD = -TCDAE*CPO                                                  AL4
      XK3VA = ZZ*CPV + LIFTVA*COSPHI                                      AL4
      XK3GA = ZZ*CPG                                                      AL4
      XK3PA = ZZ*CPP                                                      AL4
      XK3RA = ZZ*CPR + (LIFTRA - DBR*COSA)*COSPHI                         AL4
      XK3OA = ZZ*CPO                                                      AL4
      XK3MA = LIFTMA*COSPHI                                               AL4
C                    1ST PARTS W/RESP TO STATE                            AL4
   20 XK3V = LIFTV*COSPHI + YY*CPV + M*XKGV                               AL4
      XK3G = YY*CPG + M*XKGG                                              AL4
      XK3P = YY*CPP + M*XKGP                                              AL4
      XK3R = YYR*COSPHI + YY*CPR + M*XKGR                                 AL4
      XK3O = YY*CPO + M*XKGO                                              AL4
      XK3M = XKG + LIFTM*COSPHI                                           AL4
C                    2ND PARTS W/RESP TO CONTROL                          AL4
   30 XK3TD = -COSPHI*CODAE                                               AL4
      XK3TA = -XK3TD                                                      AL4
      XK3DD = -TSDAE*COSPHI                                               AL4
      XK3DA = -XK3DD                                                      AL4
      XK3AA = COSPHI*(LIFTAA + DBSA - TSDAE)                              AL4
```

```
C                      1ST PARTS W/RESP TO CONTROL                                AL4
   40 XK3T = COSPHI*SIDAE                                                         AL4
      XK3D = -TCDAE*COSPHI                                                        AL4
      XK3A = ZZ*COSPHI                                                            AL4
   50 XK3 = YY*COSPHI + M*XKG                                                     AL4
C                      CONSTRAINING EQ.                                           AL4
C                                                                                 AL4
      RETURN                                                                      AL4
      END                                                                         AL4
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol	/DYNA /(	151)	AL1	I	CODAE
						AL4	I	CODAE
						AL6	I	CODAE
						AL7	I	CODAE
						AL8	I	CODAE
						AL9	I	CODAE
						APPLY	I	CODAE
						CONTRL	I	CODAE
						NLDRV	I	CODAE
						TH3	I	CODAE
						UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol	/DYNA /(	10)	AL1	I	COSA
						AL4	I	COSA
						AL6	I	COSA
						AL7	I	COSA
						AL8	I	COSA
						AL9	I	COSA
						APPLY	I	COSA
						CONTRL	I	COSA
						NLDRV	I	COSA
						OUTPUT	I	COSA
						TH3	I	COSA
						UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol	/DYNA /(	4)	AL1	I	COSGAM
						AL4	I	COSGAM
						AL7	I	COSGAM
						AL8	I	COSGAM
						AL9	I	COSGAM
						CONTRL	I	COSGAM
						NLDRV	I	COSGAM
						OUTPUT	I	COSGAM
						PDBCOL	I	COSGAM
						STATEF	M	COSGAM
COSPHI	$\cos\phi$	I	See symbol	/DYNA /(	93)	AL1	I	COSPHI
						AL4	I	COSPHI
						APPLY	I	COSPHI
						ARCIN	O	COSPHI
						CONTRL	M	COSPHI
						OUTPUT	I	COSPHI
COSPSI	$\cos\psi$	I	See symbol	/DYNA /(	95)	AL4	I	COSPSI
						AL7	I	COSPSI
						AL8	I	COSPSI
						AL9	I	COSPSI
						CONTRL	I	COSPSI
						NLDRV	I	COSPSI
						PDBCOL	I	COSPSI
						STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol	/DYNA /(	97)	AL4	I	COSRHO
						AL7	I	COSRHO
						AL8	I	COSRHO
						AL9	I	COSRHO
						CONTRL	I	COSRHO
						NLDRV	I	COSRHO
						OUTPUT	I	COSRHO
						PDBCOL	I	COSRHO
						STATEF	M	COSRHO
COS2GM	$\cos 2\gamma$	I	See symbol	/DYNA /(	121)	AL4	I	COS2GM
						STATEF	O	COS2GM
COS2RO	$\cos 2\rho$	I	See symbol	/DYNA /(	120)	AL4	I	COS2RO
						AL7	I	COS2RO
						AL8	I	COS2RO
						NLDRV	I	COS2RO
						STATEF	O	COS2RO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DB	D_b	I	Base drag	(LBS)	/DYNA	/(163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA	/(87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
G	g	I	Instantaneous gravitational acceleration	(FT/SEC^2)	/DYNA	/(8)	AL4	I	G
							AL7	I	G
							AL8	I	G
							AL9	I	G
							CONTRL	I	G
							NLDRV	I	G
							STATEF	M	G
GAMMAD		I	Pitch rate	(RAD/SEC)	/DYNA	/(88)	AL4	I	GAMMAD
							ARCIN	O	GAMMAD
							CONTRL	I	GAMMAD
							NLDRV	I	GAMMAD
H	h	I	Integration step size in quasitime.		/D	/(2)	AL4	I	H
							INARC	M	H
							MADAMS	I	H
							RKUTT1	I	H
							RKUTT2	I	H
							SALVE	M	H
							WRAPUP	M	DT
LIFT	L	I	Aerodynamic lift	(LBS)	/DYNA	/(60)	AL4	I	LIFT
							AL5	I	LIFT
							AL6	I	LIFT
							APPLY	I	LIFT
							CONTRL	I	LIFT
							ENVPRO	I	LIFT
							OUTPUT	I	LIFT
							TH3	I	LIFT
							UT	O	LIFT
LIFTA	$\partial L/\partial \alpha$	I	See symbol		/DYNA	/(63)	AL1	I	LIFTA
							AL4	I	LIFTA
							AL5	I	LIFTA
							AL6	I	LIFTA
							APPLY	I	LIFTA
							TH3	I	LIFTA
							UT	O	LIFTA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTAA	$\partial^2 L/\partial\alpha^2$	I	See symbol	/DYNA	/(144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA
LIFTM	$\partial L/\partial m$	I	See symbol	/DYNA	/(81)	AL4	I	LIFTM
						AL5	I	LIFTM
						AL6	I	LIFTM
						APPLY	I	LIFTM
						TH3	I	LIFTM
						UT	O	LIFTM
LIFTMA	$\partial^2 L/\partial m\partial\alpha$	I	See symbol	/DYNA	/(85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA
LIFTMM	$\partial^2 L/\partial m^2$	I	See symbol	/DYNA	/(84)	AL4	I	LIFTMM
						AL5	I	LIFTMM
						AL6	I	LIFTMM
						APPLY	I	LIFTMM
						TH3	I	LIFTMM
						UT	O	LIFTMM
LIFTR	$\partial L/\partial R$	I	See symbol	/DYNA	/(62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTRA	$\partial^2 L/\partial R\partial\alpha$	I	See symbol	/DYNA	/(68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R\partial m$	I	See symbol	/DYNA	/(83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	I	See symbol	/DYNA	/(67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	I	See symbol	/DYNA	/(61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTVA	$\partial^2 L/\partial V\partial\alpha$	I	See symbol	/DYNA	/(66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA

ψv

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK		LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTVM	$\partial^2 L/\partial V \partial m$	I	See symbol		/DYNA	/(	82)	AL4	I	LIFTVM
								AL5	I	LIFTVM
								AL6	I	LIFTVM
								APPLY	I	LIFTVM
								TH3	I	LIFTVM
								UT	O	LIFTVM
LIFTVR	$\partial^2 L/\partial V \partial R$	I	See symbol		/DYNA	/(	65)	AL4	I	LIFTVR
								AL5	I	LIFTVR
								AL6	I	LIFTVR
								APPLY	I	LIFTVR
								TH3	I	LIFTVR
								UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	I	See symbol		/DYNA	/(	64)	AL4	I	LIFTVV
								AL5	I	LIFTVV
								AL6	I	LIFTVV
								APPLY	I	LIFTVV
								TH3	I	LIFTVV
								UT	O	LIFTVV
M	m	I	Mass	(G'S)	/D	/(	97)	AL4	I	M
								AL7	I	M
								AL8	I	M
								AL9	I	M
								APPLY	I	M
								BRANPT	I	M
								COSTAB	I	M
								COSTAI	I	M
								INTRPT	I	M
								NLDRV	I	M
								OUTPUT	I	M
								SALVE	I	M
								STATEF	I	M
								WRAPUP	I	M
OMEGA	ω	I	Earth rotation rate	(RAD/SEC)	/DYNA	/(	5)	AL4	I	OMEGA
								AL7	I	OMEGA
								CONTRL	I	OMEGA
								PDBCOL	I	OMEGA
								TRAJIN	M	OMEGA
OMEGA2	ω^2	I	See symbol		/DYNA	/(	6)	AL4	I	OMEGA2
								AL7	I	OMEGA2
								AL8	I	OMEGA2
								AL9	I	OMEGA2
								NLDRV	I	OMEGA2
								TRAJIN	O	OMEGA2
PG	ϕ_γ	M	See symbol		/MATS	/(	551)	AL4	M	PG
								APPLY	I	PG
								ARCIN	O	PG
								CONTRL	O	PG
PGG	$\phi_{\gamma\gamma}$	M	See symbol		/MATS	/(	560)	AL4	M	PGG
PGV	$\phi_{\gamma v}$	M	See symbol		/MATS	/(	556)	AL4	M	PGV
PO	ϕ_ρ	M	See symbol		/MATS	/(	554)	AL4	M	PO
POG	$\phi_{\rho\gamma}$	M	See symbol		/MATS	/(	563)	AL4	M	POG
POO	$\phi_{\rho\rho}$	M	See symbol		/MATS	/(	569)	AL4	M	POO
POP	$\phi_{\rho\psi}$	M	See symbol		/MATS	/(	566)	AL4	M	POP
POR	$\phi_{\rho h}$	M	See symbol		/MATS	/(	568)	AL4	M	POR
POV	$\phi_{\rho v}$	M	See symbol		/MATS	/(	559)	AL4	M	POV
PP	ϕ_ψ	M	See symbol		/MATS	/(	552)	AL4	M	PP
PPG	$\phi_{\psi\gamma}$	M	See symbol		/MATS	/(	561)	AL4	M	PPG
PPP	$\phi_{\psi\psi}$	M	See symbol		/MATS	/(	564)	AL4	M	PPP
PPV	ϕ	M	See symbol		/MATS	/(	557)	AL4	M	PPV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
PR	ϕ_h	M	See symbol	/MATS	/(553)	AL4	M	PR
PRG	$\phi_{h\gamma}$	M	See symbol	/MATS	/(562)	AL4	M	PRG
PRP	$\phi_{h\psi}$	M	See symbol	/MATS	/(565)	AL4	M	PRP
PRR	$\phi_{\lambda hh}$	M	See symbol	/MATS	/(567)	AL4	M	PRR
PRV	ϕ_{hv}	M	See symbol	/MATS	/(558)	AL4	M	PRV
PV	ϕ_v	M	See symbol	/MATS	/(550)	AL4	M	PV
PVV	ϕ_{vv}	M	See symbol	/MATS	/(555)	AL4	M	PVV
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCOL	I	R
						OLTOSZ	I	R
						STATEF	M	R
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA	/(152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA	/(9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol	/DYNA	/(3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM
SINPHI	$\sin\phi$	I	See symbol	/DYNA	/(92)	AL1	I	SINPHI
						AL4	I	SINPHI
						APPLY	I	SINPHI
						CONTRL	M	SINPHI
						OUTPUT	I	SINPHI
SINPSI	$\sin\psi$	I	See symbol	/DYNA	/(94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCOL	I	SINPSI
						STATEF	O	SINPSI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SINRHO	$\sin\rho$	I	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCOL	I	SINRHO
						STATEF	M	SINRHO
SIN2RO	$\sin 2\rho$	I	See symbol	/DYNA	/(119)	AL4	I	SIN2RO
						AL7	I	SIN2RO
						AL8	I	SIN2RO
						NLDRV	I	SIN2RO
						STATEF	M	SIN2RO
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
X	x	I	The quasitime variable.	/D	/(1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
XKG	k_γ	I	Algebraic equation used in vertical rise and pitchover	/DYNA	/(101)	AL4	I	XKG
						CONTRL	M	XKG
XKP	k_ψ	I	Algebraic equation used in vertical rise and pitchover	/DYNA	/(102)	AL4	I	XKP
						CONTRL	M	XKP

SUBRØUTINE
AL5

Purpose

AL5 evaluates the unpowered total acceleration limit constraint, Equation 10.5-1 in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL5                                                    AL5
C                                                                       AL5
C                   THIS ROUTINE APPLIES WHEN AN UNPOWERED TOTAL ACCEL. AL5
C                   LIMIT IS IN EFFECT.                                 AL5
C                                                                       AL5
C                                                                       AL5
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI     , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX     , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA     , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG     , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB     , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT       , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE      ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKB   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INDF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
```

```
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,    MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,    MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,    MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,    MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,    MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,    MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,    MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,    MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,    MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,    MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,    MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,    MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,    MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,    MATS
     *POR   ,POO   ,PLG   ,PLP                                          MATS
      EQUIVALENCE(PROD1,PROD5)                                          MATS
C                                                                       AL5
C                        THIS ENTRY COMP. EXPLICIT 2ND PARTS. W/RESP TO STATE AL5
      ENTRY AL5020                                                      AL5
      XK3VV = LIFT*LIFTVV + LIFTV**2 + DRAG*DRAGVV + DRAGV**2           AL5
      XK3VV = XK3VV + XK3VV                                             AL5
      XK3RV = LIFT*LIFTVR + LIFTV*LIFTR + DRAG*DRAGVR + DRAGV*DRAGR     AL5
      XK3RV = XK3RV + XK3RV                                             AL5
      XK3MV = LIFT*LIFTVM + LIFTV*LIFTM                                 AL5
      XK3MV = XK3MV + XK3MV                                             AL5
      XK3RR = LIFT*LIFTRR + LIFTR**2 + DRAG*DRAGRR + DRAGR**2           AL5
      XK3RR = XK3RR + XK3RR                                             AL5
      XK3MR = LIFT*LIFTRM + LIFTR*LIFTM                                 AL5
      XK3MR = XK3MR + XK3MR                                             AL5
      XK3MM = LIFT*LIFTMM + LIFTM**2 - (GMAX*GR)**2                     AL5
      XK3MM = XK3MM + XK3MM                                             AL5
C                        THIS ENTRY COMP EXPLICIT MIXED PARTS W/RESP TO STATE AL5
C                        AND CONTROL                                    AL5
      ENTRY AL5011                                                      AL5
      XK3VA = LIFT*LIFTVA + LIFTV*LIFTA + DRAG*DRAGVA + DRAGV*DRAGA     AL5
      XK3VA = XK3VA + XK3VA                                             AL5
      XK3RA = LIFT*LIFTRA + LIFTR*LIFTA + DRAG*DRAGRA + DRAGR*DRAGA     AL5
      XK3RA = XK3RA + XK3RA                                             AL5
      XK3MA = LIFT*LIFTMA + LIFTM*LIFTA                                 AL5
      XK3MA = XK3MA + XK3MA                                             AL5
C                        THIS ENTRY COMP EXPLICIT 1ST PARTS W/RESP TO STATE   AL5
      ENTRY AL5010                                                      AL5
      XK3V = LIFT*LIFTV + DRAG*DRAGV                                    AL5
      XK3V = XK3V + XK3V                                                AL5
      XK3R = LIFT*LIFTR + DRAG*DRAGR                                    AL5
      XK3R = XK3R + XK3R                                                AL5
      XK3M = LIFT*LIFTM - GR*GMAX**2*W                                  AL5
      XK3M = XK3M + XK3M                                                AL5
C                        THIS ENTRY COMP EXPLICIT 2ND PARTS W/RESP TO CONTROL AL5
      ENTRY AL5002                                                      AL5
   30 XK3AA = LIFT*LIFTAA + LIFTA**2 + DRAG*DRAGAA + DRAGA**2           AL5
      XK3AA = XK3AA + XK3AA                                             AL5
C                        THIS ENTRY COMP EXPLICIT 1ST PARTS W/RESP TO CONTROL AL5
      ENTRY AL5001                                                      AL5
   40 XK3A = LIFT*LIFTA + DRAG*DRAGA                                    AL5
      XK3A = XK3A + XK3A                                                AL5
C                        THIS ENTRY EVAL CONSTRAINING EQ. ONLY.         AL5
      ENTRY AL5000                                                      AL5
   50 TERM3 = GMAX*W                                                    AL5
      XK3 = LIFT**2 + DRAG**2 - TERM3**2                                AL5
C                                                                       AL5
      RETURN                                                            AL5
      END                                                               AL5
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA /(	69)	AL5	I	DRAG
						AL7	I	DRAG
						AL8	I	DRAG
						AL9	I	DRAG
						APPLY	I	DRAG
						CONTRL	I	DRAG
						ENVPRO	I	DRAG
						NLDRV	I	DRAG
						OUTPUT	I	DRAG
						TH3	I	DRAG
						UT	M	DRAG
DRAGA	$\partial D/\partial\alpha$	I	See symbol	/DYNA /(	72)	AL1	I	DRAGA
						AL5	I	DRAGA
						AL7	I	DRAGA
						AL8	I	DRAGA
						AL9	I	DRAGA
						APPLY	I	DRAGA
						TH3	I	DRAGA
						UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial\alpha^2$	I	See symbol	/DYNA /(	78)	AL1	I	DRAGAA
						AL5	I	DRAGAA
						AL7	I	DRAGAA
						AL8	I	DRAGAA
						AL9	I	DRAGAA
						APPLY	I	DRAGAA
						TH3	I	DRAGAA
						UT	M	DRAGAA
DRAGR	$\partial D/\partial R$	I	See symbol	/DYNA /(	71)	AL5	I	DRAGR
						AL7	I	DRAGR
						AL8	I	DRAGR
						AL9	I	DRAGR
						APPLY	I	DRAGR
						TH3	I	DRAGR
						UT	M	DRAGR
DRAGRA	$\partial^2 D/\partial R\partial\alpha$	I	See symbol	/DYNA /(	77)	AL1	I	DRAGRA
						AL5	I	DRAGRA
						AL7	I	DRAGRA
						AL8	I	DRAGRA
						AL9	I	DRAGRA
						APPLY	I	DRAGRA
						TH3	I	DRAGRA
						UT	M	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol	/DYNA /(	76)	AL5	I	DRAGRR
						AL7	I	DRAGRR
						AL8	I	DRAGRR
						AL9	I	DRAGRR
						APPLY	I	DRAGRR
						TH3	I	DRAGRR
						UT	M	DRAGRR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA /(	70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V\partial\alpha$	I	See symbol	/DYNA /(	75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA /(	74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA /(	73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV
GMAX	G_{MAX}	I	Maximum total acceleration g load	/ARCDAT/(	12)	AL5	I	GMAX
						NPLANE	I	GMAX
						THROTL	I	GMAX
						TH3	I	GMAX
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCOL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
LIFT	L	I	Aerodynamic lift (LBS)	/DYNA /(	60)	AL4	I	LIFT
						AL5	I	LIFT
						AL6	I	LIFT
						APPLY	I	LIFT
						CONTRL	I	LIFT
						ENVPRQ	I	LIFT
						OUTPUT	I	LIFT
						TH3	I	LIFT
						UT	O	LIFT
LIFTA	$\partial L/\partial \alpha$	I	See symbol	/DYNA /(	63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTAA	$\partial^2 L/\partial \alpha^2$	I	See symbol	/DYNA /(	144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA
LIFTM	$\partial L/\partial m$	I	See symbol	/DYNA /(	81)	AL4	I	LIFTM
						AL5	I	LIFTM
						AL6	I	LIFTM
						APPLY	I	LIFTM
						TH3	I	LIFTM
						UT	O	LIFTM
LIFTMA	$\partial^2 L/\partial m \partial \alpha$	I	See symbol	/DYNA /(	85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTMM	$\partial^2 L/\partial m^2$	I	See symbol	/DYNA /	(84)	AL4	I	LIFTMM
						AL5	I	LIFTMM
						AL6	I	LIFTMM
						APPLY	I	LIFTMM
						TH3	I	LIFTMM
						UT	O	LIFTMM
LIFTR	$\partial L/\partial R$	I	See symbol	/DYNA /	(62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	I	See symbol	/DYNA /	(68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R \partial m$	I	See symbol	/DYNA /	(83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	I	See symbol	/DYNA /	(67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	I	See symbol	/DYNA /	(61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	I	See symbol	/DYNA /	(66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTVM	$\partial^2 L/\partial V \partial m$	I	See symbol	/DYNA /	(82)	AL4	I	LIFTVM
						AL5	I	LIFTVM
						AL6	I	LIFTVM
						APPLY	I	LIFTVM
						TH3	I	LIFTVM
						UT	O	LIFTVM
LIFTVR	$\partial^2 L/\partial V \partial R$	I	See symbol	/DYNA /	(65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	I	See symbol	/DYNA /	(64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
W	W	I	Weight	(LBS)	/DYNA	/(91)	AL5	I	W
							ENVPRQ	I	W
							OUTPUT	I	W
							PDBCOL	I	W
							QLTOSZ	I	W
							STATEF	M	W
							TH3	I	W

SUBRØUTINE AL6

Purpose

AL6 evaluates the gravity turn constraint, Equation 10.3-1 in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL6                                                   AL6
C                                                                      AL6
C                  THIS ROUTINE APPLIES WHEN A TOTAL GRAVITY TURN IS IN AL6
C                  EFFECT                                              AL6
C                                                                      AL6
C                                                                      AL6
      LOGICAL SWITCH, ILOAD                                            DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,     DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,    DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                              DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                       DYNA
      COMMON /DYNA/                                                    DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,  JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,  DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,  DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,  DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,  DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,  DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,  DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,  DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,  DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DDRR  ,GAMMAD,AE    ,TAX   ,  DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,  DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,  DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,  DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,  DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,  DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,  DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,  DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,  DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED   DYNA
      COMMON /DYNA/                                                    DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,  DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,  DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,  DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,  DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,  DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,  JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                AUG09
      DIMENSION PROD1(2, 64)                                           MATS
      COMMON /MATS/                                                    MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,  MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,  MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,  MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,  MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,  MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,  MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,  MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,  MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,  MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,  MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,  MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,  MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,  MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT    MATS
      COMMON /MATS/                                                    MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,  MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,  MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,  MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,  MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,  MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,  MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,  MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,  MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,  MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,  MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,  MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,  MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,  MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,  MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2             MATS
      COMMON /MATS/                                                    MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
```

```
      COMMON /MATS/                                                      MATS
     *PV     ,PG     ,PP     ,PR     ,PD     ,PVV    ,PGV    ,PPV    ,PRV    ,   MATS
     *PDV    ,PGG    ,PPG    ,PRG    ,PDG    ,PPP    ,PRP    ,PDP    ,PRR    ,   MATS
     *PDR    ,PDD    ,PLG    ,PLP                                        MATS
      EQUIVALENCE(PROD1,PROD5)                                           MATS
C                                                                        AL6
C                   THIS ENTRY COMP EXPLICIT 2ND PARTS W/RESP TO STATE   AL6
      ENTRY AL6020                                                       AL6
      ASSIGN 6 TO IGO                                                    AL6
      GO TO 5                                                            AL6
C                   THIS ENTRY COMP EXPLICIT MIXED PARTS W/RESP TO STATE AL6
C                   AND CONTROL                                          AL6
      ENTRY AL6011                                                       AL6
      ASSIGN 10 TO IGO                                                   AL6
      GO TO 5                                                            AL6
C                   THIS ENTRY COMP EXPLICIT 1ST PARTS W/RESP TO STATE   AL6
      ENTRY AL6010                                                       AL6
      ASSIGN 20 TO IGO                                                   AL6
      GO TO 5                                                            AL6
C                   THIS ENTRY COMP EXPLICIT 2ND PARTS W/RESP TO CONTROL AL6
      ENTRY AL6002                                                       AL6
      ASSIGN 30 TO IGO                                                   AL6
      GO TO 5                                                            AL6
C                   THIS ENTRY COMP EXPLICIT 1ST PARTS W/RESP TO CONTROL AL6
      ENTRY AL6001                                                       AL6
      ASSIGN 40 TO IGO                                                   AL6
      GO TO 5                                                            AL6
C                   THIS ENTRY EVAL CONSTRAINING EQ.                     AL6
      ENTRY AL6000                                                       AL6
      ASSIGN 50 TO IGO                                                   AL6
C                   INITIALIZATION FOR ALL ENTRIES                       AL6
    5 TCDAE = T*CODAE                                                    AL6
      TSDAE = T*SIDAE                                                    AL6
      DBCA = DB*COSA                                                     AL6
      DBSA = DB*SINA                                                     AL6
      GO TO IGO                                                          AL6
C                   2ND PARTS W/RESP TO STATE                            AL6
    6 XK3VV = LIFTVV                                                     AL6
      XK3RV = LIFTVR                                                     AL6
      XK3MV = LIFTVM                                                     AL6
      XK3RR = LIFTRR - DBRR*SINA                                         AL6
      XK3MR = LIFTRM                                                     AL6
      XK3MM = LIFTMM                                                     AL6
C                   MIXED PARTS W/RESP TO STATE AND CONTROL              AL6
   10 XK3VA = LIFTVA                                                     AL6
      XK3RA = LIFTRA - DBR*COSA                                          AL6
      XK3MA = LIFTMA                                                     AL6
C                   1ST PARTS W/RESP TO STATE                            AL6
   20 XK3V = LIFTV                                                       AL6
      XK3R = LIFTR - DBR*SINA                                            AL6
      XK3M = LIFTM                                                       AL6
C                   2ND PARTS W/RESP TO CONTROL                          AL6
   30 XK3TD = -CODAE                                                     AL6
      XK3TA = CODAE                                                      AL6
      XK3DD = -TSDAE                                                     AL6
      XK3DA = TSDAE                                                      AL6
      XK3AA = LIFTAA + DBSA - TSDAE                                      AL6
C                   1ST PARTS W/RESP TO CONTROL                          AL6
   40 XK3T = SIDAE                                                       AL6
      XK3D = -TCDAE                                                      AL6
      XK3A = TCDAE - DBCA + LIFTA                                        AL6
C                   CONSTRAINING EQ.                                     AL6
   50 XK3 = TSDAE + LIFT - DBSA                                          AL6
C                                                                        AL6
      RETURN                                                             AL6
      END                                                                AL6
```

116

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAG SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA	/(151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA	/(10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
DB	D_b	I	Base drag	(LBS)	/DYNA	/(163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA	/(87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
LIFT	L	I	Aerodynamic lift	(LBS)	/DYNA	/(60)	AL4	I	LIFT
							AL5	I	LIFT
							AL6	I	LIFT
							APPLY	I	LIFT
							CONTRL	I	LIFT
							ENVPRO	I	LIFT
							OUTPUT	I	LIFT
							TH3	I	LIFT
							UT	O	LIFT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTA	$\partial L/\partial \alpha$	I	See symbol	/DYNA /(	63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTAA	$\partial^2 L/\partial \alpha^2$	I	See symbol	/DYNA /(	144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA
LIFTM	$\partial L/\partial m$	I	See symbol	/DYNA /(	81)	AL4	I	LIFTM
						AL5	I	LIFTM
						AL6	I	LIFTM
						APPLY	I	LIFTM
						TH3	I	LIFTM
						UT	O	LIFTM
LIFTMA	$\partial^2 L/\partial m \partial \alpha$	I	See symbol	/DYNA /(	85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA
LIFTMM	$\partial^2 L/\partial m^2$	I	See symbol	/DYNA /(	84)	AL4	I	LIFTMM
						AL5	I	LIFTMM
						AL6	I	LIFTMM
						APPLY	I	LIFTMM
						TH3	I	LIFTMM
						UT	O	LIFTMM
LIFTR	$\partial L/\partial R$	I	See symbol	/DYNA /(	62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	I	See symbol	/DYNA /(	68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R \partial m$	I	See symbol	/DYNA /(	83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	I	See symbol	/DYNA /(	67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	I	See symbol	/DYNA /(	61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	I	See symbol	/DYNA /(	66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTVM	$\partial^2 L/\partial V \partial m$	I	See symbol	/DYNA /(	82)	AL4	I	LIFTVM
						AL5	I	LIFTVM
						AL6	I	LIFTVM
						APPLY	I	LIFTVM
						TH3	I	LIFTVM
						UT	O	LIFTVM
LIFTVR	$\partial^2 L/\partial V \partial R$	I	See symbol	/DYNA /(	65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	I	See symbol	/DYNA /(	64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA /(	152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA /(	9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
T	T	I	Thrust (LBS)	/DYNA /(	42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						OL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T

SUBRØUTINE
AL7

9/15

Purpose

AL7 evaluates the dynamic pressure rate constraint, Equation 10.6-1 in Vol. I. In addition, it computes the explicit partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL7                                                    AL7
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPA  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPAM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTAM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOA  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
```

```
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1    ,PA2             MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
     *PV     ,PG     ,PP     ,PR     ,PO     ,PVV    ,PGV    ,PPV    ,PRV    ,  MATS
     *POV    ,PGG    ,PPG    ,PRG    ,POG    ,PPP    ,PRP    ,POP    ,PRR    ,  MATS
     *POR    ,POO    ,PLG    ,PLP                                       MATS
      EQUIVALENCE(PROD1,PROD5)                                          MATS
C                                                                       AL7
C         SUBROUTINE TO CALCULATE THE PARTIAL DERIVATIVES               AL7
C         OF Q-DOT AND EVALUATE A CONSTANT DYNAMIC PRESSURE             AL7
C         FLIGHT MODE CONSTRAINT                                        AL7
C                                                                       AL7
      ENTRY AL7020                                                      AL7
      ASSIGN 401 TO IGO                                                 AL7
      ASSIGN 101 TO IAM                                                 AL7
      ASSIGN 201 TO IBM                                                 AL7
      GO TO 301                                                         AL7   301
      ENTRY AL7011                                                      AL7
      ASSIGN 401 TO IGO                                                 AL7
      ASSIGN 101 TO IAM                                                 AL7
      ASSIGN 10 TO IBM                                                  AL7
      GO TO 301                                                         AL7   301
      ENTRY AL7010                                                      AL7
      ASSIGN 401 TO IGO                                                 AL7
      ASSIGN 30  TO IAM                                                 AL7
      GO TO 301                                                         AL7   301
      ENTRY AL7002                                                      AL7
      ASSIGN 40 TO IGO                                                  AL7
      GO TO 301                                                         AL7   301
      ENTRY AL7001                                                      AL7
      ASSIGN 50 TO IGO                                                  AL7
      GO TO 301                                                         AL7   301
      ENTRY AL7000                                                      AL7
      ASSIGN 60 TO IGO                                                  AL7
C                                                                       AL7
C         PRELIMINARY CALCULATIONS  ENTRY 60 ,30                        AL7
C                                                                       AL7
  301 RDOT   = V* SINGAM                                                AL7
      VDOT = R*OMEGA2*COSRHO*(COSRHO*SINGAM - SINRHO*COSPSI*COSGAM)     AL7
     * - G*SINGAM + (T*CODAE - DB*COSA - DRAG)/M                        AL7
C                                                                       AL7
C         PRELIMINARY CALC  ENTRY 50 ,40 , 10, 20                       AL7
C                                                                       AL7
  501 ROVRM = RO/M                                                      AL7
      TORO   = 2.* RO                                                   AL7
      TOROR = 2.* ROR                                                   AL7
      TORORR= 2.* RORR                                                  AL7
      GO TO IGO                                                         AL7
C                                                                       AL7
C         PRELIMINARY CALC ENTRY 30 ,10                                 AL7
C                                                                       AL7
  401 CSRHO = SIN2RO/ 2.                                                AL7
      CPSGAM= COSPSI* COSGAM                                            AL7
      CPSSGA= COSPSI* SINGAM                                            AL7
      SRHO2 = SINRHO* SINRHO                                            AL7
      CRHO2 = COSRHO* COSRHO                                            AL7
      ROMEG = R* OMEGA                                                  AL7
      ROMEG2= R* OMEGA2                                                 AL7
      CROGAM= COSRHO* COSGAM                                            AL7
      VDOTR = OMEGA2* COSRHO* ( COSRHO* SINGAM- SINRHO* CPSGAM)         AL7
     *            - GH* SINGAM        - ( DBR* COSA+ DRAGR )/ M         AL7
      VDOTV = -DRAGV/ M                                                 AL7
      RDOTV = SINGAM                                                    AL7
      RDOTG = V* COSGAM                                                 AL7
      VDOTG = ROMEG2*    COSRHO* ( CROGAM        + SINRHO* CPSSGA)      AL7
     *                 - G* COSGAM                                      AL7
      VDOTP =    ROMEG2*  CSRHO*          COSGAM* SINPSI                AL7
      VDOTO =   -ROMEG2* ( SIN2RO*SINGAM + COS2RO*CPSGAM)               AL7
      VDOTM = -( T* CODAE- DB* COSA- DRAG )/ M/ M                       AL7
      GO TO IAM                                                         AL7
C                                                                       AL7
C         PRELIMINARY CALCULATIONS TO 10                                AL7
C                                                                       AL7
  101 VDRR  = -    GRR*SINGAM        - ( DBRR* COSA+ DRAGRR )/ M        AL7
```

```
      VDRG  = OMEGA2* COSRHO* ( CROGAM        + SINRHO* CPSSGA)         AL7
     *        -     GH* COSGAM                                          AL7
      VDRP  = OMEGA2* CSRHO          * SINPSI* COSGAM                   AL7
      VDRV  = -DRAGVR/ M                                                AL7
      VDGG  = -ROMEG2   * COSRHO* ( COSRHO* SINGAM- SINRHO* CPSGAM)     AL7
     *                + G* SINGAM                                       AL7
      VDGP  = -ROMEG2   * CSRHO          * SINPSI* SINGAM               AL7
      VDPP  =  ROMEG2   * CSRHO* CPSGAM                                 AL7
      VDPO  = ROMEG2*COS2RO*SINPSI*COSGAM                               AL7
      VDOO  = -ROMEG2*2.*(COS2RO*SINGAM - SIN2RO*CPSGAM)                AL7
      VDOG  = -ROMEG2 * (SIN2RO*COSGAM - COS2RO*CPSSGA)                 AL7
      VDMM  =  2.*( T* CODAE- DB* COSA- DRAG)/M/M/M                     AL7
      VDMR  =  ( DBR* COSA+ DRAGR)/M/M                                  AL7
      VDMV  =    DRAGV/M/M                                              AL7
      GO TO IBM                                                         AL7
C-                                                                      AL7
C          PRELIMINARY CALCULATIONS ENTRY 20                            AL7
C                                                                       AL7
  201 VDOTT =  CODAE/ M                                                 AL7
      VDOTDE=  T* SIDAE/ M                                              AL7
      VDOTA = (-T* SIDAE+ DB* SINA- DRAGA )/ M                          AL7
      VDRA  = -( DRAGRA- DBR* SINA)/ M                                  AL7
      VDOTMT= -CODAE/ M/ M                                              AL7
      VDMDE = -VDOTDE/M                                                 AL7
      VDMA  =-VDOTA/M                                                   AL7
C                                                                       AL7
C          CALCULATE MIXED SECOND ORDER PARTIALS                        AL7
C                                                                       AL7
   20 XK3RT = TOROR* VDOTT                                              AL7
      XK3RD = TOROR* VDOTDE                                             AL7
      XK3RA = TOROR* VDOTA + TORO* VDRA                                 AL7
      XK3VA =-TORO* DRAGVA/M                                            AL7
      XK3MT = TORO* VDOTMT                                              AL7
      XK3MD = TORO* VDMDE                                               AL7
      XK3MA = TORO* VDMA                                                AL7
C                                                                       AL7
C          COMPUTE SECOND PARTIALS WITH RESPECT TO STATE                AL7
C                                                                       AL7
   10 XK3RR = V* RDOT* RORRR+ 2.*(RORR*VDOT+TOROR*VDOTR+RO*VDRR)        AL7
      XK3RV =    TORORR* RDOT-    TOROR* DRAGV/ M- 2.* ROVRM* DRAGVR    AL7
      XK3RG = V* RORR* RDOTG+    TOROR* VDOTG+    TORO* VDRG            AL7
      XK3RP =    TOROR* VDOTP+    TORO* VDRP                            AL7
      XK3VV =    TOROR* RDOTV- 2.*ROVRM * DRAGVV                        AL7
      XK3GV = ROR* RDOTG+ V* ROR* COSGAM                                AL7
      XK3GG = -V*V* ROR* SINGAM+    TORO* VDGG                          AL7
      XK3PG =    TORO* VDGP                                             AL7
      XK3PP =    TORO* VDPP                                             AL7
      XK3OP =    TORO* VDPO                                             AL7
      XK3OR =    TORO* VDOTO/ R+    TOROR* VDOTO                        AL7
      XK3OG =    TORO* VDOG                                             AL7
      XK3OO =    TORO* VDOO                                             AL7
      XK3MM =    TORO* VDMM                                             AL7
      XK3MR =    TORO* VDMR+    TOROR* VDOTM                            AL7
      XK3MV =    TORO* VDMV                                             AL7
C                                                                       AL7
C          FIRST PARTIALS WITH RESPECT TO STATE                         AL7
C                                                                       AL7
   30 XK3R  = V* RDOT* RORR+ TOROR* VDOT+    TORO* VDOTR                AL7
      XK3V  =    TOROR* RDOT- TORO/ M* DRAGV                            AL7
      XK3G  = V* ROR* RDOTG+ TORO* VDOTG                                AL7
      XK3P  = TORO* VDOTP                                               AL7
      XK3O = TORO*VDOTO                                                 AL7
      XK3M  = TORO* VDOTM                                               AL7
C                                                                       AL7
C                                                                       AL7
C          SECOND PARTIALS WITH RESPECT TO CONTROL                      AL7
   40 XK3TD = 2.* ROVRM* SIDAE                                          AL7
      XK3TA = -XK3TD                                                    AL7
      XK3DD = -2.* ROVRM* T* CODAE                                      AL7
      XK3DA = -XK3DD                                                    AL7
      XK3AA = -2.* ROVRM* ( T* CODAE- DB* COSA+ DRAGAA )                AL7
C                                                                       AL7
C          FIRST PARTIALS WITH RESPECT TO CONTROL                       AL7
C                                                                       AL7
   50 XK3T  = 2.* ROVRM* CODAE                                          AL7
```

```
      XK3D  = 2.* ROVRM* T* SIDAE                                     AL7
      XK3A  = 2.* ROVRM* ( DB* SINA- T* SIDAE- DRAGA )                AL7
C                                                                     AL7
C          CONSTRAINT EVALUATION                                      AL7
C                                                                     AL7
   60 XK3   = V* RDOT* ROR+ 2.* RO* VDOT                              AL7
      RETURN                                                          AL7
      END                                                             AL7
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA /(	151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA /(	10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol		/DYNA /(	4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCOL	I	COSGAM
							STATEF	M	COSGAM
COSPSI	$\cos\psi$	I	See symbol		/DYNA /(	95)	AL4	I	COSPSI
							AL7	I	COSPSI
							AL8	I	COSPSI
							AL9	I	COSPSI
							CONTRL	I	COSPSI
							NLDRV	I	COSPSI
							PDBCOL	I	COSPSI
							STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol		/DYNA /(	97)	AL4	I	COSRHO
							AL7	I	COSRHO
							AL8	I	COSRHO
							AL9	I	COSRHO
							CONTRL	I	COSRHO
							NLDRV	I	COSRHO
							OUTPUT	I	COSRHO
							PDBCOL	I	COSRHO
							STATEF	M	COSRHO
COS2RO	$\cos 2\rho$	I	See symbol		/DYNA /(	120)	AL4	I	COS2RO
							AL7	I	COS2RO
							AL8	I	COS2RO
							NLDRV	I	COS2RO
							STATEF	O	COS2RO
DB	D_b	I	Base drag	(LBS)	/DYNA /(	163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA	/(87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
DRAG	D	I	Aerodynamic drag	(LBS)	/DYNA	/(69)	AL5	I	DRAG
							AL7	I	DRAG
							AL8	I	DRAG
							AL9	I	DRAG
							APPLY	I	DRAG
							CONTRL	I	DRAG
							ENVPRO	I	DRAG
							NLDRV	I	DRAG
							OUTPUT	I	DRAG
							TH3	I	DRAG
							UT	M	DRAG
DRAGA	$\partial D/\partial \alpha$	I	See symbol		/DYNA	/(72)	AL1	I	DRAGA
							AL5	I	DRAGA
							AL7	I	DRAGA
							AL8	I	DRAGA
							AL9	I	DRAGA
							APPLY	I	DRAGA
							TH3	I	DRAGA
							UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial \alpha^2$	I	See symbol		/DYNA	/(78)	AL1	I	DRAGAA
							AL5	I	DRAGAA
							AL7	I	DRAGAA
							AL8	I	DRAGAA
							AL9	I	DRAGAA
							APPLY	I	DRAGAA
							TH3	I	DRAGAA
							UT	M	DRAGAA
DRAGR	$\partial D/\partial R$	I	See symbol		/DYNA	/(71)	AL5	I	DRAGR
							AL7	I	DRAGR
							AL8	I	DRAGR
							AL9	I	DRAGR
							APPLY	I	DRAGR
							TH3	I	DRAGR
							UT	M	DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	I	See symbol		/DYNA	/(77)	AL1	I	DRAGRA
							AL5	I	DRAGRA
							AL7	I	DRAGRA
							AL8	I	DRAGRA
							AL9	I	DRAGRA
							APPLY	I	DRAGRA
							TH3	I	DRAGRA
							UT	M	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol		/DYNA	/(76)	AL5	I	DRAGRR
							AL7	I	DRAGRR
							AL8	I	DRAGRR
							AL9	I	DRAGRR
							APPLY	I	DRAGRR
							TH3	I	DRAGRR
							UT	M	DRAGRR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA	/(70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	I	See symbol	/DYNA	/(75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA	/(74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA	/(73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV
G	g	I	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA	/(8)	AL4	I	G
						AL7	I	G
						AL8	I	G
						AL9	I	G
						CONTRL	I	G
						NLDRV	I	G
						STATEF	M	G
GH	$\partial g/\partial R$	I	See symbol	/DYNA	/(142)	AL7	I	GH
						AL8	I	GH
						NLDRV	I	GH
						STATEF	M	GH
GRR	$\partial^2 g/\partial R^2$	I	See symbol	/DYNA	/(143)	AL7	I	GRR
						AL8	I	GRR
						NLDRV	I	GRR
						STATEF	M	GRR
M	m	I	Mass (G'S)	/D	/(97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
OMEGA	ω	I	Earth rotation rate (RAD/SEC)	/DYNA	/(5)	AL4	I	OMEGA
						AL7	I	OMEGA
						CONTRL	I	OMEGA
						PDBCOL	I	OMEGA
						TRAJIN	M	OMEGA
OMEGA2	ω^2	I	See symbol	/DYNA	/(6)	AL4	I	OMEGA2
						AL7	I	OMEGA2
						AL8	I	OMEGA2
						AL9	I	OMEGA2
						NLDRV	I	OMEGA2
						TRAJIN	O	OMEGA2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA /(	7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCOL	I	R
						QLTDSZ	I	R
						STATEF	M	R
RO	ρ_a	I	Atmospheric density (SLGS/FT^3)	/DYNA /(	15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCOL	I	RO
						STATEF	I	RO
ROR	$\partial\rho_a/\partial R$	I	See symbol	/DYNA /(	19)	AL7	I	ROR
						AL8	I	ROR
						AL9	I	ROR
						NLDRV	I	ROR
						PDBCOL	I	ROR
						STATEF	I	ROR
RORR	$\partial^2\rho_a/\partial R^2$	I	See symbol	/DYNA /(	23)	AL7	I	RORR
						AL8	I	RORR
						AL9	I	RORR
						NLDRV	I	RORR
						STATEF	I	RORR
RORRR	$\partial^3\rho_a/\partial R^3$	I	See symbol	/DYNA /(	213)	AL7	I	RORRR
						AL8	I	RORRR
						AL9	I	RORRR
						STATEF	I	RORRR
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA /(	152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA /(	9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol	/DYNA /(	3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SINPSI	$\sin\psi$	I	See symbol	/DYNA	/(94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCQL	I	SINPSI
						STATEF	O	SINPSI
SINRHO	$\sin\rho$	I	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCQL	I	SINRHO
						STATEF	M	SINRHO
SIN2RO	$\sin 2\rho$	I	See symbol	/DYNA	/(119)	AL4	I	SIN2RO
						AL7	I	SIN2RO
						AL8	I	SIN2RO
						NLDRV	I	SIN2RO
						STATEF	M	SIN2RO
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCQL	I	V
						STATEF	I	V
						WRAPUP	I	V

SUBRØUTINE AL8

926

Purpose

AL8 evaluates the heating rate constraint, Equation 10.6-2 in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL8                                                    AL8
C                                                                       AL8
C         SUBROUTINE TO CALCULATE THE PARTIAL DERIVATIVES               AL8
C         OF Q-DOT-DOT  AND EVALUATE A CONSTANT HEATING                 AL8
C         RATE FLIGHT MODE CONSTRAINT                                   AL8
C                                                                       AL8
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   J
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
```

```
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,      MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,      MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,      MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,      MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,      MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                 MATS
      COMMON /MATS/                                                        MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)     MATS
      COMMON /MATS/                                                        MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,      MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,      MATS
     *POR   ,POO   ,PLG   ,PLP                                             MATS
      EQUIVALENCE(PROD1,PROD5)                                             MATS
      ENTRY AL8020                                                         AL8
      ASSIGN 201 TO IAM                                                    AL8
      ASSIGN  20 TO IGO                                                    AL8
      GO TO  601                                                           AL8   601
      ENTRY AL8011                                                         AL8
      ASSIGN 101 TO IAM                                                    AL8
      ASSIGN 10  TO IGO                                                    AL8
      GO TO 601                                                            AL8   601
      ENTRY AL8010                                                         AL8
      ASSIGN 301 TO IAM                                                    AL8
      ASSIGN 301 TO IGO                                                    AL8
      GO TO 601                                                            AL8   601
      ENTRY AL8002                                                         AL8
      ASSIGN 401 TO IAM                                                    AL8
      ASSIGN 40 TO IGO                                                     AL8
      GO TO 601                                                            AL8   601
      ENTRY AL8001                                                         AL8
      ASSIGN 501 TO IAM                                                    AL8
      ASSIGN 50 TO IGO                                                     AL8
      GO TO 601                                                            AL8   601
      ENTRY AL8000                                                         AL8
      ASSIGN 60 TO IAM                                                     AL8
C                                                                          AL8
C          PRELIMINARY CALCULATIONS FOR ENTRY 60                           AL8
C                                                                          AL8
  601 ROOT   = V* SINGAM                                                   AL8
      R6R    = ROR/6.3                                                     AL8
      R6RR   = RORR/6.3                                                    AL8
      R6RRR  = RORRR/6.3                                                   AL8
      V115   = V**1.15                                                     AL8
      CSRHO  = SIN2RO /2.                                                  AL8
      CPSGAM = COSPSI* COSGAM                                              AL8
      CPSSGA = COSPSI* SINGAM                                              AL8
      ROMEG2 = R* OMEGA2                                                   AL8
      CROGAM = COSRHO* COSGAM                                              AL8
      CROSGA = COSRHO* SINGAM                                              AL8
      VDOT   = ROMEG2   * COSRHO*( CROSGA-         SINRHO* CPSGAM)         AL8
     1                 - G* SINGAM+ (T* CODAE- DB* COSA- DRAG)/ M          AL8
      GO TO IAM                                                            AL8
C                                                                          AL8
C          PRELIMINARY CALCULATIONS FOR ENTRY 20                           AL8
C                                                                          AL8
  201 VDRA   = (DBR* SINA - DRAGRA)/ M                                     AL8
      VDVA   = - DRAGVA/ M                                                 AL8
      VDMT   = - CODAE/M/M                                                 AL8
      VDMD   = - T* SIDAE/M/M                                              AL8
      VDMA   =  (T* SIDAE- DB* SINA + DRAGA)/M/M                           AL8
C                                                                          AL8
C          PRELIMINARY CALCULATIONS FOR ENTRY 10                           AL8
C                                                                          AL8
  101 VDRR   = -   GRR       *SINGAM - (DBRR* COSA + DRAGRR)/M             AL8
      VDRV   = -DRAGVR/M                                                   AL8
      VDRG   = OMEGA2* COSRHO*(CROGAM + SINRHO* CPSSGA)- GH*COSGAM         AL8
      VDRP   = OMEGA2* CSRHO* SINPSI*COSGAM                                AL8
      VDRO   = -OMEGA2*( SIN2RO* SINGAM + COS2RO* CPSGAM)                  AL8
      VDRM   = (DBR* COSA + DRAGR)/M/M                                     AL8
      VDVV   = -DRAGVV/M                                                   AL8
      VDVM   =  DRAGV/M/M                                                  AL8
      VDGG   = -ROMEG2* COSRHO*(CROSGA-SINRHO*CPSGAM)+ G* SINGAM           AL8
      VDGP   = -R* VDRP*SINGAM/COSGAM                                      AL8
      VDGD   = - ROMEG2 *(SIN2RO * COSGAM - COS2RO*CPSSGA)                 AL8
      VDPO   = ROMEG2 * COSGAM* SINPSI* COS2RO                             AL8
```

```
      VDPP = ROMEG2*CSRHO*CPSGAM                                          AL8
      VDOO = -2.*ROMEG2*(COS2RO*SINGAM - SIN2RO*CPSGAM)                   AL8
      VDMM = 2.*(T*CODAE - DB*COSA - DRAG)/M**3                           AL8
C                                                                         AL8
C         PRELIMINARY CALCULATIONS FOR ENTRY 30 ,20 ,10                   AL8
C                                                                         AL8
  301 VDOTR  = OMEGA2* COSRHO*( CROSGA - SINRHO* CPSGAM) -                AL8
     1           GH* SINGAM        - ( DBR* COSA + DRAGR)/ M              AL8
      RDOTV  = SINGAM                                                     AL8
      VDOTV  =-DRAGV/ M                                                   AL8
      RDOTG  = V* COSGAM                                                  AL8
      VDOTG  = ROMEG2* COSRHO*( CROGAM+ SINRHO* CPSSGA)- G* COSGAM        AL8
      VDOTP  = ROMEG2* CSRHO * SINPSI* COSGAM                             AL8
      VDOTO  =-ROMEG2*( 2.*CSRHO * SINGAM + COS2RO* CPSGAM)               AL8
      VDOTM  =-(T*CODAE - DB* COSA - DRAG)/M/M                            AL8
C                                                                         AL8
C         PRELIMINARY CALCULATIONS FOR ENTRY 40                           AL8
C                                                                         AL8
  401 VDTD   = SIDAE/M                                                    AL8
      VDTA   = - VDTD                                                     AL8
      VDDD   = - T*CODAE/M                                                AL8
      VDDA   = - VDDD                                                     AL8
      VDAA   =      VDDD   -( DB* COSA + DRAGAA)/ M                       AL8
C                                                                         AL8
C         PRELIMINARY CALCULATIONS FOR ENTRY 50                           AL8
C                                                                         AL8
  501 VDOTT  = CODAE/M                                                    AL8
      VDOTD  = T* SIDAE/ M                                                AL8
      VDOTA  =   -VDOTD   +(DB* SINA - DRAGA)/M                           AL8
      GO TO 160                                                           AL8
C                                                                         AL8
C         CALCULATE  MIXED PARTIALS                                       AL8
C                                                                         AL8
   20 XK3RT  = ROR* VDOTT                                                 AL8
      XK3RD  = ROR* VDOTD                                                 AL8
      XK3RA  = ROR* VDOTA+  RO* VDRA                                      AL8
      XK3VA  = RO * VDVA                                                  AL8
      XK3MT  = RO * VDMT                                                  AL8
      XK3MD  = RO * VDMD                                                  AL8
      XK3MA  = RO * VDMA                                                  AL8
C                                                                         AL8
C         CALCULATE SECOND PARTIALS WITH RESPECT TO STATE                 AL8
C                                                                         AL8
   10 XK3RR  = RORR* VDOT+ 2.*ROR* VDOTR+ RO* VDRR+V115*R6RRR*RDOT        AL8
      XK3RV  = ROR * VDOTV+ RO* VDRV+R6RR*V115*SINGAM*2.15                AL8
      XK3RG  = ROR * VDOTG+ RO* VDRG+R6RR*V115*RDOTG                      AL8
      XK3RP  = ROR * VDOTP+ RO* VDRP                                      AL8
      XK3OR  = ROR * VDOTO+ RO* VDRO                                      AL8
      XK3MR  = ROR * VDOTM+ RO* VDRM                                      AL8
      XK3VV =2.4725*SINGAM*R6R*V115/V+RO*VDVV                             AL8
      XK3GV  =  V115*R6R*2.15*COSGAM                                      AL8
      XK3MV  = RO* VDVM                                                   AL8
      XK3GG  =-V115*R6R*RDOT + RO* VDGG                                   AL8
      XK3PG  = RO* VDGP                                                   AL8
      XK3OG  = RO* VDGO                                                   AL8
      XK3OP  = RO* VDPO                                                   AL8
      XK3PP = RO*VDPP                                                     AL8
      XK3OO = RO*VDOO                                                     AL8
      XK3MM = RO*VDMM                                                     AL8
C                                                                         AL8
C         CALCULATE FIRST PARTIALS WITH RESPECT TO STATE                  AL8
C                                                                         AL8
   30 XK3R   = ROR* VDOT+ RO* VDOTR+V115*R6RR*RDOT                        AL8
      XK3V   = V115*R6R*2.15*SINGAM+RO*VDOTV                              AL8
      XK3G   =  V115*R6R*RDOTG + RO*VDOTG                                 AL8
      XK3P   = RO* VDOTP                                                  AL8
      XK3O   = RO* VDOTO                                                  AL8
      XK3M   = RO* VDOTM                                                  AL8
C                                                                         AL8
C         CALCULATE SECOND PARTIALS WITH RESPECT TO CONTROL               AL8
C                                                                         AL8
   40 XK3TD  =  RO*VDTD                                                   AL8
      XK3TA  =  RO*VDTA                                                   AL8
      XK3DD  =  RO*VDDD                                                   AL8
      XK3DA  =  RO*VDDA                                                   AL8
```

```
      XK3AA  =  RO*VDAA                                                   AL8
C                                                                         AL8
C           CALCULATE FIRST PARTIALS WITH RESPECT TO CONTROL              AL8
C                                                                         AL8
   50 XK3T   = RO* VDOTT                                                  AL8
      XK3D   = RO* VDOTD                                                  AL8
      XK3A   = RO* VDOTA                                                  AL8
C                                                                         AL8
C           CALCULATE CONSTRAINT                                          AL8
C                                                                         AL8
   60 XK3    = VI15*RDOT*R6R + RO*VDOT                                    AL8
      RETURN                                                              AL8
      END                                                                 AL8
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol	/DYNA	/(151)	AL1	I	CODAE
						AL4	I	CODAE
						AL6	I	CODAE
						AL7	I	CODAE
						AL8	I	CODAE
						AL9	I	CODAE
						APPLY	I	CODAE
						CONTRL	I	CODAE
						NLDRV	I	CODAE
						TH3	I	CODAE
						UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol	/DYNA	/(10)	AL1	I	COSA
						AL4	I	COSA
						AL6	I	COSA
						AL7	I	COSA
						AL8	I	COSA
						AL9	I	COSA
						APPLY	I	COSA
						CONTRL	I	COSA
						NLDRV	I	COSA
						OUTPUT	I	COSA
						TH3	I	COSA
						UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol	/DYNA	/(4)	AL1	I	COSGAM
						AL4	I	COSGAM
						AL7	I	COSGAM
						AL8	I	COSGAM
						AL9	I	COSGAM
						CONTRL	I	COSGAM
						NLDRV	I	COSGAM
						OUTPUT	I	COSGAM
						PDBCOL	I	COSGAM
						STATEF	M	COSGAM
COSPSI	$\cos\psi$	I	See symbol	/DYNA	/(95)	AL4	I	COSPSI
						AL7	I	COSPSI
						AL8	I	COSPSI
						AL9	I	COSPSI
						CONTRL	I	COSPSI
						NLDRV	I	COSPSI
						PDBCOL	I	COSPSI
						STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol	/DYNA	/(97)	AL4	I	COSRHO
						AL7	I	COSRHO
						AL8	I	COSRHO
						AL9	I	COSRHO
						CONTRL	I	COSRHO
						NLDRV	I	COSRHO
						OUTPUT	I	COSRHO
						PDBCOL	I	COSRHO
						STATEF	M	COSRHO
COS2RO	$\cos 2\rho$	I	See symbol	/DYNA	/(120)	AL4	I	COS2RO
						AL7	I	COS2RO
						AL8	I	COS2RO
						NLDRV	I	COS2RO
						STATEF	O	COS2RO
DB	D_b	I	Base drag (LBS)	/DYNA	/(163)	AL1	I	DB
						AL4	I	DB
						AL6	I	DB
						AL7	I	DB
						AL8	I	DB
						AL9	I	DB
						APPLY	I	DB
						CONTRL	I	DB
						NLDRV	I	DB
						OUTPUT	I	DB
						STATEF	I	DB
						TH3	I	DB
						UT	I	DB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DBR	$\partial D_b/\partial R$	I	See symbol	/DYNA /(	86)	AL1	I	DBR
						AL4	I	DBR
						AL6	I	DBR
						AL7	I	DBR
						AL8	I	DBR
						AL9	I	DBR
						APPLY	I	DBR
						STATEF	I	DBR
						TH3	I	DBR
						UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol	/DYNA /(	87)	AL4	I	DBRR
						AL6	I	DBRR
						AL7	I	DBRR
						AL8	I	DBRR
						AL9	I	DBRR
						APPLY	I	DBRR
						STATEF	I	DBRR
						TH3	I	DBRR
						UT	I	DBRR
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA /(	69)	AL5	I	DRAG
						AL7	I	DRAG
						AL8	I	DRAG
						AL9	I	DRAG
						APPLY	I	DRAG
						CONTRL	I	DRAG
						ENVPRO	I	DRAG
						NLDRV	I	DRAG
						OUTPUT	I	DRAG
						TH3	I	DRAG
						UT	M	DRAG
DRAGA	$\partial D/\partial \alpha$	I	See symbol	/DYNA /(	72)	AL1	I	DRAGA
						AL5	I	DRAGA
						AL7	I	DRAGA
						AL8	I	DRAGA
						AL9	I	DRAGA
						APPLY	I	DRAGA
						TH3	I	DRAGA
						UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial \alpha^2$	I	See symbol	/DYNA /(	78)	AL1	I	DRAGAA
						AL5	I	DRAGAA
						AL7	I	DRAGAA
						AL8	I	DRAGAA
						AL9	I	DRAGAA
						APPLY	I	DRAGAA
						TH3	I	DRAGAA
						UT	M	DRAGAA
DRAGR	$\partial D/\partial R$	I	See symbol	/DYNA /(	71)	AL5	I	DRAGR
						AL7	I	DRAGR
						AL8	I	DRAGR
						AL9	I	DRAGR
						APPLY	I	DRAGR
						TH3	I	DRAGR
						UT	M	DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	I	See symbol	/DYNA /(	77)	AL1	I	DRAGRA
						AL5	I	DRAGRA
						AL7	I	DRAGRA
						AL8	I	DRAGRA
						AL9	I	DRAGRA
						APPLY	I	DRAGRA
						TH3	I	DRAGRA
						UT	M	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol	/DYNA /(	76)	AL5	I	DRAGRR
						AL7	I	DRAGRR
						AL8	I	DRAGRR
						AL9	I	DRAGRR
						APPLY	I	DRAGRR
						TH3	I	DRAGRR
						UT	M	DRAGRR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA /(	70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	I	See symbol	/DYNA /(	75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA /(	74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA /(	73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV
G	g	I	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA /(	8)	AL4	I	G
						AL7	I	G
						AL8	I	G
						AL9	I	G
						CONTRL	I	G
						NLDRV	I	G
						STATEF	M	G
GH	$\partial g/\partial R$	I	See symbol	/DYNA /(	142)	AL7	I	GH
						AL8	I	GH
						NLDRV	I	GH
						STATEF	M	GH
GRR	$\partial^2 g/\partial R^2$	I	See symbol	/DYNA /(	143)	AL7	I	GRR
						AL8	I	GRR
						NLDRV	I	GRR
						STATEF	M	GRR
M	m	I	Mass (G'S)	/D /(	97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
OMEGA2	ω^2	I	See symbol	/DYNA /(	6)	AL4	I	OMEGA2
						AL7	I	OMEGA2
						AL8	I	OMEGA2
						AL9	I	OMEGA2
						NLDRV	I	OMEGA2
						TRAJIN	O	OMEGA2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCOL	I	R
						OLTOSZ	I	R
						STATEF	M	R
RO	ρ_a	I	Atmospheric density ($SLGS/FT^3$)	/DYNA	/(15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCOL	I	RO
						STATEF	I	RO
ROR	$\partial\rho_a/\partial R$	I	See symbol	/DYNA	/(19)	AL7	I	ROR
						AL8	I	ROR
						AL9	I	ROR
						NLDRV	I	ROR
						PDBCOL	I	ROR
						STATEF	I	ROR
RORR	$\partial^2\rho_a/\partial R^2$	I	See symbol	/DYNA	/(23)	AL7	I	RORR
						AL8	I	RORR
						AL9	I	RORR
						NLDRV	I	RORR
						STATEF	I	RORR
RORRR	$\partial^3\rho_a/\partial R^3$	I	See symbol	/DYNA	/(213)	AL7	I	RORRR
						AL8	I	RORRR
						AL9	I	RORRR
						STATEF	I	RORRR
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA	/(152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA	/(9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol	/DYNA	/(3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM

435

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBR	CODE	VAR
SINPSI	$\sin\psi$	I	See symbol		/DYNA /(	94)	AL4	I	SINPSI
							AL7	I	SINPSI
							AL8	I	SINPSI
							AL9	I	SINPSI
							CONTRL	I	SINPSI
							NLDRV	I	SINPSI
							PDBCOL	I	SINPSI
							STATEF	O	SINPSI
SINRHO	$\sin\rho$	I	See symbol		/DYNA /(	96)	AL4	I	SINRHO
							AL7	I	SINRHO
							AL8	I	SINRHO
							AL9	I	SINRHO
							CONTRL	I	SINRHO
							NLDRV	I	SINRHO
							OUTPUT	I	SINRHO
							PDBCOL	I	SINRHO
							STATEF	M	SINRHO
SIN2RO	$\sin 2\rho$	I	See symbol		/DYNA /(	119)	AL4	I	SIN2RO
							AL7	I	SIN2RO
							AL8	I	SIN2RO
							NLDRV	I	SIN2RO
							STATEF	M	SIN2RO
T	T	I	Thrust	(LBS)	/DYNA /(	42)	ALGCON	M	T
							AL1	I	T
							AL4	I	T
							AL6	I	T
							AL7	I	T
							AL8	I	T
							AL9	I	T
							APPLY	I	T
							ARCIN	O	T
							CONTRL	M	T
							DL2	I	T
							IMPULS	I	T
							OUTPUT	I	T
							TH1	I	T
							TH2	I	T
							TH3	I	T
							TH4	I	T
V	V	I	Relative velocity.	(FT/SEC)	/D /(	91)	AL1	I	V
							AL4	I	V
							AL7	I	V
							AL8	I	V
							AL9	I	V
							BCOND	I	NOM
							BNDRY	O	NOM
							BRANPT	M	NOM
							CONTRL	I	V
							ENDPT	I	NOM
							ENVPRO	I	V
							FETCH	O	NOM
							INTERP	M	V
							INTRPT	M	NOM
							NLDRV	O	NOM
							NLDRV	I	V
							OUTPUT	I	V
							PDBCOL	I	V
							STATEF	I	V
							WRAPUP	I	V

SUBRØUTINE AL9

Purpose

AL9 evaluates the Reynolds number rate constraint, Equation 10.6-3 in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE AL9                                                    AL9
C                                                                       AL9
C         SUBROUTINE TO CALCULATE THE PARTIAL DERIVATIVES               AL9
C         OF REYNOLDS NUMBER-DOT AND EVALUATE A CONSTANT                AL9
C         REYNOLDS NUMBER FLIGHT MODE CONSTRAINT                        AL9
C                                                                       AL9
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CBOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
```

```
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,      MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,      MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,      MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,      MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,      MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                 MATS
      COMMON /MATS/                                                        MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)    MATS
      COMMON /MATS/                                                        MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV    ,     MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR    ,     MATS
     *POR   ,POO   ,PLG   ,PLP                                             MATS
      EQUIVALENCE(PROD1,PROD5)                                             MATS
C                                                                          AL9
      ENTRY AL9020                                                         AL9
      ASSIGN 10 TO IAM                                                     AL9
      ASSIGN  1 TO IGO                                                     AL9
      GO TO 6                                                              AL9
      ENTRY AL9011                                                         AL9
      ASSIGN 20 TO IAM                                                     AL9
      ASSIGN  2 TO IGO                                                     AL9
      GO TO 6                                                              AL9
      ENTRY AL9010                                                         AL9
      ASSIGN 30 TO IAM                                                     AL9
      ASSIGN  3 TO IGO                                                     AL9
      GO TO 6                                                              AL9
      ENTRY AL9002                                                         AL9
      ASSIGN 40 TO IAM                                                     AL9
      ASSIGN  4 TO IGO                                                     AL9
      GO TO 6                                                              AL9
      ENTRY AL9001                                                         AL9
      ASSIGN 50 TO IAM                                                     AL9
      ASSIGN  5 TO IGO                                                     AL9
      GO TO 6                                                              AL9
      ENTRY AL9000                                                         AL9
      ASSIGN 60 TO IGO                                                     AL9
      GO TO 6                                                              AL9
C                                                                          AL9
    6 RDOT   = V* SINGAM                                                   AL9
      VDOT   = R* OMEGA2* COSRHO*(COSRHO* SINGAM- SINRHO* COSPSI* COSGAM AL9
     *         )- G* SINGAM+ (T* CODAE- DB* COSA- DRAG)/ M                 AL9
      C1     = RHO* UMU/ V                                                 AL9
      C2     = UMU* ROR - UMU1* RO                                         AL9
      C3     = C2/ V                                                       AL9
      GO TO IGO                                                            AL9
C                                                                          AL9
    1 CONTINUE                                                             AL9
      VDOTT  = CODAE/M                                                     AL9
      VDVA   = - DRAGVA/M                                                  AL9
      VDRA   = -(DRAGRA - DBR* SINA)/M                                     AL9
      F1     =(ROR* UMU + UMU1*RO)/V                                       AL9
C                                                                          AL9
    2 VDRR   =-3.*G/R/R* SINGAM- (DRAGRR+ DBRR*COSA)/ M                    AL9
      VDRG   = OMEGA2* COSRHO*(COSRHO* SINGAM+ SINRHO* COSPSI* SINGAM)+    AL9
     *         G*COSGAM/R                                                  AL9
      VDRP   = OMEGA2* COSRHO* SINRHO* SINPSI* COSGAM                      AL9
      B1     = COSRHO* SINGAM -SINRHO* COSPSI* COSGAM                      AL9
      B2     =-SINRHO* SINGAM -COSRHO* COSPSI* COSGAM                      AL9
      VDRG   =-OMEGA2*(SINRHO* B1+ COSRHO* B2)                             AL9
      VDRM   = (DRAGR + DBR*COSA)/M/M                                      AL9
      VDVV   = - DRAGVV/ M                                                 AL9
      VDGG   = R* OMEGA2* COSRHO* SINRHO* COSPSI* COSGAM + G* SINGAM       AL9
      VDGP   =-R* OMEGA2* COSRHO* SINRHO* SINPSI* SINGAM                   AL9
      B3     = COSRHO* COSGAM + SINRHO* COSPSI* SINGAM                     AL9
      B4     = SINRHO* COSGAM - COSRHO* COSPSI* SINGAM                     AL9
      VDGO   =-R* OMEGA2*(SINRHO* B3 + COSRHO* B4)                         AL9
      VDPP   =-R* OMEGA2* COSRHO* SINRHO* COSPSI* COSGAM                   AL9
      B22    = SINRHO* SINGAM - COSRHO* COSPSI* COSGAM                     AL9
      VDOO   =-R*OMEGA2*(2.*COSRHO*B1-SINRHO*(B22- B2))                    AL9
      VDMM   = 2.*( T* CODAE- DB*COSA -DRAG)/M/M/M                         AL9
C                                                                          AL9
    3 A1     = COSRHO* SINGAM- SINRHO* COSPSI* COSGAM                      AL9
      A2     = COSRHO* COSGAM+ SINRHO* COSPSI* SINGAM                      AL9
      A3     = SINRHO* SINGAM+ COSRHO* COSPSI* COSGAM                      AL9
      VDOTR  = OMEGA2* COSRHO* A1+ G* SINGAM/R -(DRAGR+ DBR* COSA)/M       AL9
```

```
      VDOTV  = - DRAGV/ M                                               AL9
      VDOTG  = R* OMEGA2* COSGAM* A2 - G* COSGAM                        AL9
      VDOTP  =-R* OMEGA2*(A1+ A3)                                       AL9
      VDOTM  =-(T* CODAE- DB* COSA - DRAG)/ M                           AL9
C                                                                       AL9
    4 CONTINUE                                                          AL9
C                                                                       AL9
    5 C4     = C1/M                                                     AL9
      GO TO IAM                                                         AL9
C                                                                       AL9
C           CALCULATE MIXED PARTIALS                                    AL9
C                                                                       AL9
   10 XK3RT  = VDOTT*F1                                                 AL9
      XK3RD  = VDOTD*F1                                                 AL9
      XK3RA  = VDOTA*F1 + C1* VDRA                                      AL9
      XK3VT  = -C1/V* VDOTT                                             AL9
      XK3VD  = -C1/V* VDOTD                                             AL9
      XK3VA  = -C1/V* VDOTA + C1* VDVA                                  AL9
      XK3MT  =-C4    * VDOTT                                            AL9
      XK3MD  =-C4    * VDOTD                                            AL9
      XK3MA  =-C4    * VDOTA                                            AL9
C                                                                       AL9
C           CALCULATE SECOND PARTIALS WITH RESPECT TO STATE             AL9
C                                                                       AL9
   20 XK3RR  = RDOT*(UMU*RORRR+UMU1*RORR-ROR*UMU2-RO*UMU3)+ VDOT/V*(    AL9
     *          2.*ROR*UMU1 +RORR*UMU +UMU2*RO)+ C1*VDRR                AL9
      XK3RV  =(UMU*RORR-RO*UMU2)*SINGAM -VDOT/V*(ROR*UMU + UMU1*RO)+    AL9
     *          (ROR*UMU+UMU1*RO)/V/M*VDOTV +C1*(DRAGVR/M -  VDOTR/V)   AL9
      XK3RG  =-C2/V* VDOTG -V*COSGAM*(UMU*RORR-RO*UMU2)                 AL9
      XK3RP  = C1*VDRP + VDOTP*C2/V                                     AL9
      XK3RO  = C1*VDRO + VDOTO*C2/V                                     AL9
      XK3RM  = C1*VDRM + VDOTM*C2/V                                     AL9
      XK3VV  = C1*VDVV- 2.*C1/V/V* VDOTV + C1/V/V/V* VDOT               AL9
      XK3VG  = C2*COSGAM - C1/V*VDOTG                                   AL9
      XK3VP  =-C1/V*VDOTP                                               AL9
      XK3VO  =-C1/V*VDOTO                                               AL9
      XK3VM  =-C1/V*VDOTM    +  C4* DRAGV                               AL9
      XK3GG  =-C2* V* SINGAM + C1* VDGG                                 AL9
      XK3GP  = C1* VDGP                                                 AL9
      XK3GO  = C1* VDGO                                                 AL9
      XK3PP  = C1* VDPP                                                 AL9
      XK3PO  = C1* VDPP                                                 AL9
      XK3OO  = C1* VDOO                                                 AL9
      XK3MM  = C1* VDMM                                                 AL9
C                                                                       AL9
C           CALCULATE FIRST PARTIALS WITH RESPECT TO STATE              AL9
C                                                                       AL9
   30 XK3R   = RDOT*(UMU*RORR-RO*UMU1)+VDOT/V*(ROR*UMU+UMU1*RO)+C1*VDOTR AL9
      XK3V   = C2* SINGAM - C1/V* VDOT + C1* VDOTV                      AL9
      XK3G   = C2* V* COSGAM+ C1* VDOTG                                 AL9
      XK3P   = C1* VDOTP                                                AL9
      XK3O   = C1* VDOTO                                                AL9
      XK3M   = C1* VDOTM                                                AL9
C                                                                       AL9
C           CALCULATE SECOND PARTIALS WITH RESPECT TO CONTROL           AL9
C                                                                       AL9
   40 XK3TD  = C4* SIDAE                                                AL9
      XK3TA  = - XK3TD                                                  AL9
      XK3DA  = C4* T*CODAE                                              AL9
      XK3DD  = - XK3DA                                                  AL9
      XK3AA  = C4* (DB* COSA - T* CODAE- DRAGAA)                        AL9
C                                                                       AL9
C           CALCULATE FIRST PARTIALS WITH RESPECT TO CONTROL            AL9
C                                                                       AL9
   50 XK3T   = C4    * CODAE                                            AL9
      XK3D   = C4    * T* SIDAE                                         AL9
      XK3A   = C4    * (DB* SINA- T* SIDAE- DRAGA)                      AL9
C                                                                       AL9
C           CALCULATE CONSTRAINT                                        AL9
C                                                                       AL9
   60 X53    = C2* RDOT + C1* VDOT                                      AL9
      RETURN                                                            AL9
      END                                                               AL9
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA	/(151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA	/(10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol		/DYNA	/(4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCOL	I	COSGAM
							STATEF	M	COSGAM
COSPSI	$\cos\psi$	I	See symbol		/DYNA	/(95)	AL4	I	COSPSI
							AL7	I	COSPSI
							AL8	I	COSPSI
							AL9	I	COSPSI
							CONTRL	I	COSPSI
							NLDRV	I	COSPSI
							PDBCOL	I	COSPSI
							STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol		/DYNA	/(97)	AL4	I	COSRHO
							AL7	I	COSRHO
							AL8	I	COSRHO
							AL9	I	COSRHO
							CONTRL	I	COSRHO
							NLDRV	I	COSRHO
							OUTPUT	I	COSRHO
							PDBCOL	I	COSRHO
							STATEF	M	COSRHO
DB	D_b	I	Base drag	(LBS)	/DYNA	/(163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA	/(87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
DRAG	D	I	Aerodynamic drag	(LBS)	/DYNA	/(69)	AL5	I	DRAG
							AL7	I	DRAG
							AL8	I	DRAG
							AL9	I	DRAG
							APPLY	I	DRAG
							CONTRL	I	DRAG
							ENVPRO	I	DRAG
							NLDRV	I	DRAG
							OUTPUT	I	DRAG
							TH3	I	DRAG
							UT	A	DRAG
DRAGA	$\partial D/\partial \alpha$	I	See symbol		/DYNA	/(72)	AL1	I	DRAGA
							AL5	I	DRAGA
							AL7	I	DRAGA
							AL8	I	DRAGA
							AL9	I	DRAGA
							APPLY	I	DRAGA
							TH3	I	DRAGA
							UT	A	DRAGA
DRAGAA	$\partial^2 D/\partial \alpha^2$	I	See symbol		/DYNA	/(78)	AL1	I	DRAGAA
							AL5	I	DRAGAA
							AL7	I	DRAGAA
							AL8	I	DRAGAA
							AL9	I	DRAGAA
							APPLY	I	DRAGAA
							TH3	I	DRAGAA
							UT	A	DRAGAA
DRAGR	$\partial D/\partial R$	I	See symbol		/DYNA	/(71)	AL5	I	DRAGR
							AL7	I	DRAGR
							AL8	I	DRAGR
							AL9	I	DRAGR
							APPLY	I	DRAGR
							TH3	I	DRAGR
							UT	A	DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	I	See symbol		/DYNA	/(77)	AL1	I	DRAGRA
							AL5	I	DRAGRA
							AL7	I	DRAGRA
							AL8	I	DRAGRA
							AL9	I	DRAGRA
							APPLY	I	DRAGRA
							TH3	I	DRAGRA
							UT	A	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol		/DYNA	/(76)	AL5	I	DRAGRR
							AL7	I	DRAGRR
							AL8	I	DRAGRR
							AL9	I	DRAGRR
							APPLY	I	DRAGRR
							TH3	I	DRAGRR
							UT	A	DRAGRR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA	/(70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	I	See symbol	/DYNA	/(75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA	/(74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA	/(73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV
G	g	I	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA	/(8)	AL4	I	G
						AL7	I	G
						AL8	I	G
						AL9	I	G
						CONTRL	I	G
						NLDRV	I	G
						STATEF	M	G
M	m	I	Mass (G'S)	/D	/(97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
OMEGA2	ω^2	I	See symbol	/DYNA	/(6)	AL4	I	OMEGA2
						AL7	I	OMEGA2
						AL8	I	OMEGA2
						AL9	I	OMEGA2
						NLDRV	I	OMEGA2
						TRAJIN	O	OMEGA2
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCOL	I	R
						QLTOSZ	I	R
						STATEF	M	R

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
RHO	ρ	I	Latitude	(RAD)	/D	/(95)	AL9	I	RHO
							OUTPUT	I	RHO
							STATEF	I	RHO
							WRAPUP	I	RHO
RO	ρ_a	I	Atmospheric density	($SLGS/FT^3$)	/DYNA	/(15)	AL7	I	RO
							AL8	I	RO
							AL9	I	RO
							NLDRV	I	RO
							OUTPUT	I	RO
							PDBCOL	I	RO
							STATEF	I	RO
ROR	$\partial\rho_a/\partial R$	I	See symbol		/DYNA	/(19)	AL7	I	ROR
							AL8	I	ROR
							AL9	I	ROR
							NLDRV	I	ROR
							PDBCOL	I	ROR
							STATEF	I	ROR
RORR	$\partial^2\rho_a/\partial R^2$	I	See symbol		/DYNA	/(23)	AL7	I	RORR
							AL8	I	RORR
							AL9	I	RORR
							NLDRV	I	RORR
							STATEF	I	RORR
RORRR	$\partial^3\rho_a/\partial R^3$	I	See symbol		/DYNA	/(213)	AL7	I	RORRR
							AL8	I	RORRR
							AL9	I	RORRR
							STATEF	I	RORRR
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol		/DYNA	/(152)	AL1	I	SIDAE
							AL4	I	SIDAE
							AL6	I	SIDAE
							AL7	I	SIDAE
							AL8	I	SIDAE
							AL9	I	SIDAE
							APPLY	I	SIDAE
							CONTRL	I	SIDAE
							TH3	I	SIDAE
							UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol		/DYNA	/(9)	AL1	I	SINA
							AL4	I	SINA
							AL6	I	SINA
							AL7	I	SINA
							AL8	I	SINA
							AL9	I	SINA
							APPLY	I	SINA
							CONTRL	I	SINA
							OUTPUT	I	SINA
							TH3	I	SINA
							UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol		/DYNA	/(3)	AL1	I	SINGAM
							AL4	I	SINGAM
							AL7	I	SINGAM
							AL8	I	SINGAM
							AL9	I	SINGAM
							CONTRL	I	SINGAM
							NLDRV	I	SINGAM
							PDBCOL	I	SINGAM
							STATEF	M	SINGAM
SINPSI	$\sin\psi$	I	See symbol		/DYNA	/(94)	AL4	I	SINPSI
							AL7	I	SINPSI
							AL8	I	SINPSI
							AL9	I	SINPSI
							CONTRL	I	SINPSI
							NLDRV	I	SINPSI
							PDBCOL	I	SINPSI
							STATEF	O	SINPSI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SINRHO	$\sin\rho$	I	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCOL	I	SINRHO
						STATEF	M	SINRHO
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V

Purpose

ANL62S computes the atmospheric properties T_a, P_a, ρ_a and c_s for the '62 Standard atmosphere. Moreover, when the Reynolds number rate constraint is in effect, it computes the atmospheric viscosity ν. In addition, the following derivatives are computed:

$$\frac{dT_a}{dh}, \frac{dP_a}{dh}, \frac{d\rho_a}{dh} \text{ and } \frac{dc_s}{dh}$$

and

$$\frac{d^2T_a}{dh^2}, \frac{d^2P_a}{dh^2}, \frac{d^2\rho_a}{dh^2} \text{ and } \frac{d^2c_s}{dh^2}$$

and, if any SVIC's are in effect

$$\frac{d^3\rho_a}{dh^3}$$

and, if a Reynolds number rate constraint is in effect

$$\frac{d\nu}{dh}, \frac{d^3\nu}{\partial h^2} \text{ and } \frac{d^3\nu}{dh^3}$$

```
      SUBROUTINE ANL62S(H,DIM,DOM,IDAM)                                  ANL62S
C                                                                        ANL62S
      DIMENSION DIM(12)                                                  ANL62S
      DIMENSION  DOM(5)                                                  ANL62S
      DATA A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15,A16,A17,   ANL62S
     *A18,A19,A20,A21,A22,A23,A24,A25,A26,A27,A28,A29,A30,A31,A32        ANL62S
     *   / -.14655396E-6,6356.77,.25653341E-10,6356.77,.14116834E-3,     ANL62S
     *   14.002385,-.38282910E-4,216.23225,.15084978E-3,26.414270,       ANL62S
     *   684.10967,.67419880E-3,.044294588,.58500460,.85519675E-4,       ANL62S
     *   137.47450,10533.544,.49863416E-4,.013120767,.90188546,          ANL62S
     *  -.25392354E-3,-193.32352,10180.367,.11921879E-2,.034567717,      ANL62S
     *  -3.3413764,-.33888604E-4,-384.32662,38131.516,.89812379E-4,      ANL62S
     *   .028810210,-5.5362354/                                          ANL62S
      DATA B9,B8,B7,B6,B5,B4,B3,B2,B1,B0                                 ANL62S
     */ .12840084E  2,-.25307047E  0, .22460528E- 2,-.11672987E- 4,      ANL62S
     *  .38922542E- 7,-.86034597E-10, .12577373E-12,-.11709468E-15,      ANL62S
     *   .62944405E-19,-.14878777E-22/                                   ANL62S
      DATA  CN/ -3483.67635/                                             ANL62S
      DATA C10,C9,C8,C7,C6,C5,C4,C3,C2,C1,C0                             ANL62S
     *   / 283.7492391, -3.955242007,-.5232974573, .05256403630,         ANL62S
     *   -.1832962145E-2, .3432295909E-4, -.3930824139E-6, .2848535349E  ANL62S
     *   -8, -.1269919974E-10, .3161762924E-13, -.3350145769E-16/        ANL62S
      DATA D3,D2,D1,D0                                                   ANL62S
     *   / 253.316666, 6.8049727, -.56317016E-2,  .12742813E-5 /         ANL62S
      DATA E9,E8,E7,E6,E5,E4,E3,E2,E1,E0                                 ANL62S
     */ .47255200E  2,-.96148320E  0, .87302363E- 2,-.46139476E- 4,      ANL62S
     *  .15580020E- 6,-.34772427E- 9, .51217501E-12,-.47967392E-15,      ANL62S
     *  .25907821E-18,-.61476982E-22/                                    ANL62S
      DATA      CC8,CC7,CC6,CC5,CC4,CC3,CC2,CC1,CC0                      ANL62S
     *  /                -1.0465949146,.1576921089,-.733184858E-2,       ANL62S
     *      1.7161479545E-4,-2.3584944834E-6,1.9939747443E-8,            ANL62S
     *     -1.0159359792E-10,2.8455866316E-13,-3.350145769E-16/          ANL62S
      DATA       CCC7,CCC6,CCC5,CCC4,CCC3,CCC2,CCC1,CCC0                 ANL62S
     *  /                .3153842178,-2.199554574E-2,6.864591818E-4,     ANL62S
     *     -1.1792472417E-5,1.1963848465BE-7,-7.1115518544E-10,          ANL62S
     *      2.2764693052BE-12,-3.0151311921E-15/                         ANL62S
      DATA  CCCC6,CCCC5,CCCC4,CCCC3,CCCC2,CCCC1,CCCC0                    ANL62S
     *     /-4.399109148E-2,2.0593775454E-3,-4.7169889668E-5,            ANL62S
     *     5.9819242329E-7,-4.2669311264E-9,1.5935285136688E-11,         ANL62S
     *     -2.41210495368E-14/                                           ANL62S
      DATA ROWXTO /3.5298375E11/                                         ANL62S
      DATA       DD1,DD0                                                 ANL62S
     *  /              -1.12634032E-2,.38228439E-5/                      ANL62S
      DATA       DDD0                                                    ANL62S
     *  /                 .76456878E-5/                                  ANL62S
      DATA      BB7,BB6,BB5,BB4,BB3,BB2,BB1,BB0                          ANL62S
     *  /                 .44921056E-2,-.35018961E-4,1.55690168E-7,      ANL62S
     *     -4.30172985E-10,.75464238E-12,-.81966276E-15,5.0355524E-19,   ANL62S
     *     -1.33908993E-22/                                              ANL62S
      DATA       BBB6,BBB5,BBB4,BBB3,BBB2,BBB1,BBB0                      ANL62S
     *  /                  -.70037922E-4,4.67070504E-7,-1.72069194E-9,   ANL62S
     *      3.7732119E-12,-4.91797656E-15,3.52488668E-18,                ANL62S
     *     -1.071271944E-21/                                             ANL62S
      DATA      EE7,EE6,EE5,EE4,EE3,EE2,EE1,EE0                          ANL62S
     *  /              1.74604726E-2,-1.38418428E-4,.6232008E-6,         ANL62S
     *     -1.73862135E-9,3.07305006E-12,-3.35771744E-15,2.07262568E-18, ANL62S
     *     -5.53292838E-22/                                              ANL62S
      DATA       EEE6,EEE5,EEE4,EEE3,EEE2,EEE1,EEE0                      ANL62S
     *  /                  -2.76836856E-4,1.8696024E-6,-6.9544854E-9,    ANL62S
     *      1.53652503E-11,-2.014630464E-14,1.4508379 76E-17,            ANL62S
     *     -4.426342704E-21/                                             ANL62S
      DATA  EEEE5,EEEE4,EEEE3,EEEE2,EEEE1,EEEE0                          ANL62S
     *      /3.7392048E-6,-2.08634562E-8,6.14610012E-11,-1.007315232E-13, ANL62S
     *      8.705027856E-17,-3.0984398928E-20/                           ANL62S
      DATA CON, HAFCON                                                   ANL62S
     */2.004680277637 86E-2,1.00234013881893E-2/                         ANL62S
      DATA POWERO, PO/-.28016066E-2,.101325E9/                           ANL62S
      DATA X2,X3,X4,X5,X6,X7,X9,X11,X12                                  ANL62S
     */2.0885400839168 93E-5,1.9403172295773 53E-12,3.2808398950131 23E3, ANL62S
     *5.4864E-4,6.3658701757786 57E-9,5.9140869157517 31E-16,            ANL62S
     *1.67225472E-7,1.8026136919211 16E-19,3.048E-4/                     ANL62S
      DATA  COEF1,SC,TF,UMUF,UMUF2,UMUF3,TFSC/                           ANL62S
     *     1.7818E-4, 198., 392., 3.02E-7, 9.1204E-14, 27.543608E-21,    ANL62S
     * 590./                                                             ANL62S
      POWER(Z) = A1/(Z + A2) + A3*ALOG(Z + A4) + A5*ALOG(Z + A6) +       ANL62S
```

```
     *              A7*ALOG(A8 - Z) + A9*ALOG(Z*(Z - A10) + A11) +         ANL62S
     *              A12*ATAN(A13*Z - A14) + A15*ALOG(Z*(Z - A16) + A17) +  ANL62S
     *              A18*ATAN(A19*Z - A20) + A21*ALOG(Z*(Z + A22) + A23) +  ANL62S
     *              A24*ATAN(A25*Z + A26) + A27*ALOG(Z*(Z + A28) + A29) +  ANL62S
     *              A30*ATAN(A31*Z + A32)                                  ANL62S
C                                                                          ANL62S
      IF(IDAM.GT.0.0) GO TO 1                                              ANL62S
      IF(IDAM.LT.0.0) GO TO 2                                              ANL62S
      ASSIGN 50 TO IGO                                                     ANL62S
      GO TO 40                                                             ANL62S
    1 ASSIGN 20 TO IAM                                                     ANL62S
      ASSIGN 10 TO IGO                                                     ANL62S
      GO TO 40                                                             ANL62S
    2 ASSIGN 50 TO IAM                                                     ANL62S
      ASSIGN 10 TO IGO                                                     ANL62S
   40 X =  H * X12                                                         ANL62S
      IF (X .GT. 550.)  X=550.                                             ANL62S
      IF (X .LT. 0.   )  X= 0.                                             ANL62S
      IF ( X.GT. 195. )   GO TO 1000                                       ANL62S
C                                                                          ANL62S
C     SET  CONSTANTS                                                       ANL62S
      POWERX= POWER(X)                                                     ANL62S
C                                                                          ANL62S
CTM                                                                        ANL62S
      TM = C10 + X*(C9 + X*(C8 + X*(C7 + X*(C6 + X*(C5 + X*(C4 + X*(C3 + ANL62S
     *     X*(C2 + X*(C1 + X*C0)))))))))                                   ANL62S
      TM1 =  C9 + X*(CC8 + X*(CC7 + X*(CC6 + X*(CC5 + X*(CC4 + X*(CC3 +   ANL62S
     *       X*(CC2 + X*(CC1 + X*CC0))))))))                               ANL62S
      TM2 =  CC8 + X*(CCC7 + X*(CCC6 + X*(CCC5 + X*(CCC4 + X*(CCC3 +      ANL62S
     *       X*(CCC2 + X*(CCC1 + X*CCC0)))))))                             ANL62S
      TM3 = CCC7 + X*(CCCC6 + X*(CCCC5 + X*(CCCC4 + X*(CCCC3 + X*(CCCC2   ANL62S
     *       + X*(CCCC1 + X*CCCC0))))))                                    ANL62S
C                                                                          ANL62S
C GRAVITY PARTIALS                                                         ANL62S
      R   =  6356.765 + X                                                  ANL62S
      G   =  .396271577E6/R**2                                             ANL62S
      G1  =  -2.*G/R                                                       ANL62S
      G2  =  -3.*G1/R                                                      ANL62S
C G/TM PARTIALS                                                            ANL62S
      GTM =   G/TM                                                         ANL62S
      GTM1=  (G1*TM - G*TM1)/ TM**2                                        ANL62S
      GTM2 = (TM1*(TM*(G2 - G1 - G1) + 2.*G*TM1) - G*TM*TM2)/TM**3         ANL62S
C EXPONET                                                                  ANL62S
      EXPO=. EXP( CN*(POWERX -  POWERO) )                                  ANL62S
      EXP1=  EXPO*CN*GTM                                                   ANL62S
      EXP2 = CN*(EXP1*GTM + EXPO*GTM1)                                     ANL62S
      EXP3= CN*(2.* EXP1* GTM1 + GTM* EXP2+ EXPO* GTM2)                    ANL62S
C PRESSURE                                                                 ANL62S
      P   =  PO* EXPO                                                      ANL62S
      P1  =  PO* EXP1                                                      ANL62S
      P2  =  PO* EXP2                                                      ANL62S
C DENSITY                                                                  ANL62S
      RHO =  ROWXTO*EXPO/TM                                                ANL62S
      RHO1= (ROWXTO*EXP1 - RHO*TM1)/TM                                     ANL62S
      RHO2= (ROWXTO*EXP2-2.*RHO1*TM1-RHO*TM2)/TM                           ANL62S
C                                                                          ANL62S
      GO TO IGO                                                            ANL62S
C                                                                          ANL62S
 1000 CONTINUE                                                             ANL62S
C                                                                          ANL62S
C  TM .GT. 195                                                             ANL62S
      TM = D3 + X*(D2 + X*(D1 + X*D0))                                     ANL62S
      TM1 =  D2 + X*(DD1 + X*DD0) .                                        ANL62S
      TM2 =  DD1 + X*DDD0                                                  ANL62S
C                                                                          ANL62S
C ROW .GT. 195                                                             ANL62S
C   P  .GT. 195                                                            ANL62S
      RHO = E9 + X*(E8 + X*(E7 + X*(E6 + X*(E5 + X*(E4 + X*(E3 + X*(E2 + ANL62S
     *      X*(E1 + X*E0))))))))                                           ANL62S
      P = B9 + X*(B8 + X*(B7 + X*(B6 + X*(B5 + X*(B4 + X*(B3 + X*(B2 +    ANL62S
     *    X*(B1 + X*B0))))))))                                             ANL62S
C                                                                          ANL62S
      RHO1 =  E8 + X*(EE7 + X*(EE6 + X*(EE5 + X*(EE4 + X*(EE3 + X*(EE2 + ANL62S
```

```
      *       X*(EE1 + X*EE0)))))))                                   ANL62S
       P1 =  B8 + X*(BB7 + X*(BB6 + X*(BB5 + X*(BB4 + X*(BB3 + X*(BB2 +  ANL62S
      *     X*(BB1 + X*BB0)))))))                                     ANL62S
C                                                                     ANL62S
       RHO2 =  EE7 + X*(EEE6 + X*(EEE5 + X*(EEE4 + X*(EEE3 + X*(EEE2 +  ANL62S
      *       X*(EEE1 + X*EEE0))))))                                  ANL62S
       P2 =  BB7 + X*(BBB6 + X*(BBB5 + X*(BBB4 + X*(BBB3 + X*(BBB2 +    ANL62S
      *     X*(BBB1 + X*BBB0))))))                                    ANL62S
       GO TO 160                                                      ANL62S
C                                                                     ANL62S
C                                                                     ANL62S
C CS                                                                  ANL62S
   10 CONTINUE                                                        ANL62S
      IF(X.GT.195.) GO TO 5                                           ANL62S
      RHO3= (ROWXTO* EXP3 - 3.*(RHO1* TM2+ RHO2* TM1)- RHO* TM3)/TM    ANL62S
      GO TO 7                                                         ANL62S
    5 RHO3 = EEE6 + X*(EEEE5 + X*(EEEE4 + X*(EEEE3 + X*(EEEE2 + X*(EEEE1 ANL62S
     *          + X* EEEE0)))))                                       ANL62S
C                                                                     ANL62S
C          DYNAMIC VISCOSITY CALCULATIONS                             ANL62S
C                                                                     ANL62S
    7 TTF32  = (1.8*TM/TF)*SQRT(1.8*TM/TF)                            ANL62S
      TSC    = 1.8*TM + SC                                            ANL62S
      UMU    = COEF1* TTF32/TSC                                       ANL62S
      GO TO IAM                                                       ANL62S
   20 CONTINUE                                                        ANL62S
      TM      = 1.8* TM                                               ANL62S
      TM1     = X5 * TM1                                              ANL62S
      TM2     = X9 * TM2                                              ANL62S
      TM3     = X9/X4*TM3                                             ANL62S
      TM20    = TM*TM                                                 ANL62S
      TM30    = TM20*TM                                               ANL62S
      GT      = TFSC/ TSC                                             ANL62S
      GT1     = -GT/ TSC                                              ANL62S
      GT2     = 2.* GT/ TSC/ TSC                                      ANL62S
      GT3     =-2.* GT/ TSC/TSC/TSC                                   ANL62S
      FT      = TTF32                                                 ANL62S
      FT1     = 1.5* FT/ TM                                           ANL62S
      FT2     = +0.75* FT/ TM20                                       ANL62S
      FT3     =-0.375* FT/ TM30                                       ANL62S
      UMU1    = UMUF*(FT*GT1*TM1 + GT*FT1*TM1)                        ANL62S
      UMU2    = UMUF*((FT*GT1+ GT*FT1)* TM2 +                         ANL62S
     *          (FT*GT2 + 2.*FT1*GT1 + GT*FT2)*TM1*TM1)               ANL62S
      UMU3    = UMUF*((FT*GT1 + GT*FT1)*TM3 +                         ANL62S
     *          (FT*GT2 + 2.*FT1*GT1 + GT*FT2)* 3.*TM1*TM2 +          ANL62S
     *          (FT*GT3 + 3.*FT1*GT2 + 3.*GT1*FT2 + GT*FT3)*TM1*TM1*TM1)  ANL62S
      TM      = TM/1.8                                                ANL62S
      TM1     = TM1/X5                                                ANL62S
      TM2     = TM2/X9                                                ANL62S
      TM3     = TM3/X9*X4                                             ANL62S
   50 SQRTTM = SQRT(TM)                                               ANL62S
      CS = CON*SQRTTM                                                 ANL62S
      CS1 = HAFCON*TM1/SQRTTM                                         ANL62S
      CS2 = HAFCON*(TM*TM2 - .5*TM1**2)/(TM*SQRTTM)                   ANL62S
C                                                                     ANL62S
 1500 DIM( 1) = 1.8*TM                                                ANL62S
      DIM( 2) = X2*P                                                  ANL62S
      DIM( 3) = X3*RHO                                                ANL62S
      DIM( 4) = X4*CS                                                 ANL62S
      DIM( 5) = X5*TM1                                                ANL62S
      DIM( 6) = X6*P1                                                 ANL62S
      DIM( 7) = X7*RHO1                                               ANL62S
      DIM( 8) = CS1                                                   ANL62S
      DIM( 9) = X9*TM2                                                ANL62S
      DIM(10) = X3*P2                                                 ANL62S
      DIM(11) = X11*RHO2                                              ANL62S
      DIM(12) =X12*CS2                                                ANL62S
      DOM(1) = RHO3 * X11/X4                                          ANL62S
      DOM(2) =  UMU                                                   ANL62S
      DOM(3) =  UMU1                                                  ANL62S
      DOM(4) =  UMU2                                                  ANL62S
      DOM(5) =  UMU3                                                  ANL62S
      RETURN                                                          ANL62S
      END                                                             ANL62S
```

Purpose

ANL63P computes the atmospheric properties T_a, P_a, ρ_a and c_s for the '63 Patrick A.F.B. atmosphere. Moreover, when the Reynolds number rate constraint is in effect, it computes the atmospheric viscosity ν. In addition, the following derivatives

$$\frac{dT_a}{dh}, \frac{dP_a}{dh}, \frac{d\rho_a}{dh} \text{ and } \frac{dc_s}{dh}$$

and

$$\frac{d^2T_a}{dh^2}, \frac{d^2P_a}{dh^2}, \frac{d^2\rho_a}{dh^2} \text{ and } \frac{d^2c_s}{dh^2}$$

and, if any SVIC's are in effect

$$\frac{d^3\rho_a}{dh^3}$$

and, if a Reynolds number rate constraint is in effect

$$\frac{d\nu}{dh}, \frac{d^2\nu}{dh^2} \text{ and } \frac{d^3\nu}{dh^3}$$

```
      SUBROUTINE ANL63P(Q,PV,DOM,IDAM)                                   ANL63P
      DIMENSION PV(12)                                                   ANL63P
      DIMENSION  DOM(5)                                                  ANL63P
      DATA  R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13 /              ANL63P
     *           8.3080902734182425, 3.6752156174349436        ,         ANL63P
     *            .5487134541479621 5, 1.4678068127507852       ,         ANL63P
     *          -.6146337871757466 8, -1.5399754331174728      ,         ANL63P
     *           .8266841570529131 1,  1.2852428316497672      ,         ANL63P
     *          -.5723450117375682 6, -.6223801281528296 1     ,         ANL63P
     *           .1722984003852595 8,  .1480957679862631 3     ,         ANL63P
     *          -.0183353102935381 99 ,-.0134982767300404 8   /         ANL63P
      DATA  S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13 /              ANL63P
     *           8.49701849953499 9    ,4.1680512787110 44    ,          ANL63P
     *           1.001136979720595   , .375689209583064 4   ,            ANL63P
     *          -1.287997408735040   , 0.014970860707497 03,             ANL63P
     *          1.674006844535595   , .130034680585550 8 ,               ANL63P
     *          -1.176664525367179   , -.248063347106903 ,               ANL63P
     *           .364055154432592 3   , .106274166038221 3 ,             ANL63P
     *          -.039861819263869 85  , -.013645479194274 45 /           ANL63P
      DATA  T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13 /              ANL63P
     *             -.10248437640715577   , -.20389894971168512 ,         ANL63P
     *             -.02039938067127985 2 , .33088492304959274   ,        ANL63P
     *             -.5547318590645693 9  , -.55482818087304384  ,        ANL63P
     *              .9859565556903839 8  , .67846024779392942   ,        ANL63P
     *             -.6667767437849189 6  ,-.40978650717250266   ,        ANL63P
     *              .1968859121466848 6  , .11368136922321633   ,        ANL63P
     *            -0.02090139046687915 5 ,-0.01159595019181811 2 /       ANL63P
      DATA  U0,U1,U2,U3,U4,U5,U6,U7,U8,U9,U10,U11,U12 /                  ANL63P
     *3.67521561743494 ,  1.09742690829592 ,  4.40342043825234 ,         ANL63P
     *-2.45853514810299, -7.69987716558737 ,  4.96010494231746 ,         ANL63P
     *8.99669982154836 , -4.57876009390054 , -5.60142115337547 ,         ANL63P
     *1.72298400385259 ,  1.62905344784890 , -.220023723522458 ,         ANL63P
     *-.175477597490526/                                                 ANL63P
      DATA  V0,V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,V12 /                  ANL63P
     *4.16805127871103 ,  2.00227395944118 ,  1.12706762874920 ,         ANL63P
     *-5.15198963494015, 0.0748543035374852,  10.0440410672136 ,         ANL63P
     *.910242764098854 , -9.41331620293744 , -2.23257012396212 ,         ANL63P
     *3.64055154432592 ,  1.16901582642043 , -.478341831166437 ,         ANL63P
     *-.17739122952557 /                                                 ANL63P
      DATA  W0,W1,W2,W3,W4,W5,W6,W7,W8,W9,W10,W11,W12/                   ANL63P
     *-.203898949711685, -.0407987613425598, .992654769148778   ,        ANL63P
     *-2.21892743625828, -2.77414090436523 , 5.91573933414227   ,        ANL63P
     *4.74922173455749 , -5.33421395027935 ,-3.68807856455253   ,        ANL63P
     *1.96885912146685 ,  1.25049506145537 ,-0.250816685602551 ,         ANL63P
     *-0.150747352493635/                                                ANL63P
      DATA  X0,X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11 /                      ANL63P
     *1.09742690829592 ,  8.80684087650468 , -7.37560544610898 ,         ANL63P
     *-30.7995086623495, 24.8005247115872 9 , 53.9801989292900 5 ,       ANL63P
     *-32.0513206573037,-44.8113692270037 5 , 15.5068560346733 2 ,       ANL63P
     *16.2905344784888 9, -2.42026095874704  , -2.10573116988631 /       ANL63P
      DATA  Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11 /                      ANL63P
     *2.00227395944118 ,  2.25413525749839 ,-15.4559689048204 5 ,        ANL63P
     *0.299417214149941, 50.2202053360679 2 ,  5.46145658459312 ,        ANL63P
     *-65.893213420562 ,-17.8605609916969 6 , 32.7649638989330 5 ,       ANL63P
     *11.6901582642043 , -5.26176014283081 , -2.12869475430681 /         ANL63P
      DATA  Z0,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11 /                      ANL63P
     *-.0407987613425598, 1.98530953829756 , -6.65678230877484 ,         ANL63P
     *-11.0965636174609 , 29.5786966707113 4 , 28.4953304073449 1 ,      ANL63P
     *-37.3394976519553 ,-29.5046285164202 1 , 17.7197320932016 3 ,      ANL63P
     *12.5049506145537 1 , -2.75898354162806 , -1.80896822992361 /       ANL63P
      DATA  RHO, P0,A0 /.00229645    , 2124.0827,1137.98  /              ANL63P
      DATA   B1,B2,B3,B4,B5,B6,B7,B8,B9,B10 /                            ANL63P
     *-14.7512108922179 6, -92.3985259870485 , 99.2020988463491 6,       ANL63P
     *269.900994647450 25, -192.307923943822 2, -313.679584589026 25,    ANL63P
     *124.054848277386 56, 146.614810306400 01, -24.2026095874704,       ANL63P
     *-23.1630428687494 1/                                               ANL63P
      DATA   CONST1, CONST2/                                             ANL63P
     * 4.161533415E-4, 8.323066830E-4/                                   ANL63P
      DATA COEF1,SC,TF,UMUF,TFSC/                                        ANL63P
     *     1.7818E-4, 198., 392., 3.02E-7, 590./                         ANL63P
      DATA A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10/                  ANL63P
     *1.98530953829756,    -13.3135646175496 8, -33.2896908523827 0,     ANL63P
     *118..147866828453 6, 142.4766520367245 5, -224.0369859117318,      ANL63P
     *-206.532399614941 47,141.757856745613 04,  112.544555530992 39,    ANL63P
     *-27.5898354162806,  -19.8986505291597 1/                           ANL63P
```

```
      IF(IDAM.GT.0) GO TO 1                                             ANL63P
      IF(IDAM.LT.0) GO TO 2                                             ANL63P
      ASSIGN 50 TO IGO                                                  ANL63P
      GO TO 40                                                          ANL63P
    1 ASSIGN 20 TO IAM                                                  ANL63P
      ASSIGN 10 TO IGO                                                  ANL63P
      GO TO 40                                                          ANL63P
    2 ASSIGN 50 TO IAM                                                  ANL63P
      ASSIGN 10 TO IGO                                                  ANL63P
   40 H = Q                                                             ANL63P
      IF(H.LT.0.) H=0.                                                  ANL63P
      X= (H- 2.E5)*1.E-5                                                ANL63P
C                                                                       ANL63P
C          CALCULATE  DENSITY, PRESSURE, AND SPEED OF SOUND             ANL63P
C                                                                       ANL63P
      PXRHO = R0 + X*(R1+ X*(R2 +X*(R3 +X*(R4 +X*(R5 +X*(R6 +X*(R7 +X*( ANL63P
     *  R8 +X*(R9 +X*(R10+X*(R11+X*(R12 +X*R13))))))))))))              ANL63P
      PXP  = S0 + X*(S1+X *(S2 +X*(S3 +X* (S4 +X*(S5 +X*(S6 +X*(S7 +X*( ANL63P
     *  S8 +X*(S9 +X*(S10+X*(S11+X*(S12 +X*S13))))))))))))              ANL63P
      PXA  = T0 + X*(T1+X *(T2 +X*(T3 +X* (T4 +X*(T5 +X*(T6 +X*(T7 +X*( ANL63P
     *  T8 +X*(T9 +X*(T10+X*(T11+X*(T12 +X*T13))))))))))))              ANL63P
      P = P0*EXP(-PXP)                                                  ANL63P
      RHO = RHO*EXP(-PXRHO)                                             ANL63P
      A = A0*EXP(PXA)                                                   ANL63P
      PV(2) = P                                                         ANL63P
      PV(3) = RHO                                                       ANL63P
      PV(4) = A                                                         ANL63P
C                                                                       ANL63P
C          CALCULATE FIRST DERIVATIVES WO RADIUS                        ANL63P
C                                                                       ANL63P
      DXRHO = U0 +X*(U1 +X*(U2+X*(U3+X*(U4+X*(U5+X*(U6 +X*(U7 +X*(U8+   ANL63P
     *  X*(U9 +X*(U10 +X*(U11+X*U12)))))))))))                          ANL63P
      DXP  = V0 + X*(V1 +X*(V2+X*(V3+X*(V4+X*(V5+X*(V6 +X*(V7 +X*(V8+   ANL63P
     *  X*(V9 +X*(V10 +X*(V11+X*V12)))))))))))                          ANL63P
      DXA  = W0 + X*(W1 +X*(W2+X*(W3+X*(W4+X*(W5+X*(W6 +X*(W7 +X*(W8+   ANL63P
     *  X*(W9 +X*(W10 +X*(W11+X*W12)))))))))))                          ANL63P
      PV(6) = -1.E-5*P*DXP                                              ANL63P
      PV(7) = -1.E-5*RHO*DXRHO                                          ANL63P
      PV(8) =  1.E-5*A*DXA                                              ANL63P
C                                                                       ANL63P
C          CALCULATE SECOND DERIVATIVES WO RADIUS                       ANL63P
C                                                                       ANL63P
      QXRHO = X0 + X*(X1 + X*(X2 + X*(X3 + X*(X4 + X*(X5 + X*(X6 + X*(X7 ANL63P
     * + X*(X8 + X*(X9 + X*(X10 + X*X11)))))))))))                      ANL63P
      QXP   = Y0 + X*(Y1 + X*(Y2 + X*(Y3 + X*(Y4 + X*(Y5 + X*(Y6 + X*(Y7 ANL63P
     * + X*(Y8 + X*(Y9 + X*(Y10 + X*Y11)))))))))))                      ANL63P
      QXA   = Z0 + X*(Z1 + X*(Z2 + X*(Z3 + X*(Z4 + X*(Z5 + X*(Z6 + X*(Z7 ANL63P
     * + X*(Z8 + X*(Z9 + X*(Z10 + X*Z11)))))))))))                      ANL63P
      PV(10)= -1.E-10*P*(QXP - DXP**2)                                  ANL63P
      PV(11)= -1.E-10*RHO*(QXRHO - DXRHO**2)                            ANL63P
      PV(12)=  1.E-10*A*(QXA + DXA**2)                                  ANL63P
      GO TO IGO                                                         ANL63P
   10 CONTINUE                                                          ANL63P
C                                                                       ANL63P
C          CALCULATE THIRD DERIVATIVES WO RADIUS                        ANL63P
C                                                                       ANL63P
      SXRHO = X1+ X*(B1 + X*(B2 + X*(B3 + X*(B4 + X*(B5 + X*(B6 +       ANL63P
     *          X*(B7 + X*(B8 + X*(B9 + X* B10)))))))))                 ANL63P
      SXA    = A0 + X*(A1 + X*(A2 + X*(A3 + X*(A4 + X*(A5 + X*(A6 +     ANL63P
     *           X*(A7 + X*(A8 + X*(A9 + X*A10)))))))))                 ANL63P
      DOM(1) = -1.E-15* RHO*(SXRHO- (3.*QXRHO- DXRHO*DXRHO)*DXRHO)      ANL63P
      ARRR = 1.E-15* A*(SXA + (3.*QXA + DXA*DXA)*DXA)                   ANL63P
C                                                                       ANL63P
C          TEMPERATURE CALCULATIONS                                     ANL63P
C                                                                       ANL63P
      TM      =    A *    A * CONST1                                    ANL63P
C                                                                       ANL63P
C          DYNAMIC VISCOSITY CALCULATIONS                               ANL63P
C                                                                       ANL63P
      FT   = (TM/TF)* SQRT(TM/TF)                                       ANL63P
      TSC =  TM + SC                                                    ANL63P
      UMU =   COEF1 * FT/TSC                                            ANL63P
      DOM(2) = UMU                                                      JUL21
      GO TO IAM                                                         ANL63P
```

```
      20 CONTINUE                                                         ANL63P
C                                                                         ANL63P
C           CALCULATE TEMPERATURE AND FIRST THREE DERIVATIVES WO RADIUS  ANL63P
C                                                                         ANL63P
         TM1     = A* PV(8)* CONST2                                       ANL63P
         TM2     = CONST2 *(A* PV(12) + PV(8)* PV(8))                     ANL63P
         TM3     = CONST2 * (A* ARRR + 3.* PV(12)* PV(8))                 ANL63P
         TM20 = TM*TM                                                     ANL63P
         TM30 = TM20* TM                                                  ANL63P
         GT   = TFSC/TSC                                                  ANL63P
         GT1  = -GT /TSC                                                  ANL63P
         GT2  = 2.* GT/ TSC/TSC                                           ANL63P
         GT3  =- GT2/TSC                                                  ANL63P
         FT1  = 1.5* FT/TM                                                ANL63P
         FT2  = 0.75* FT/TM20                                             ANL63P
         FT3  =-0.375* FT/TM30                                            ANL63P
         UMU1 = UMUF*(FT* GT1 + GT* FT1)* TM1                             ANL63P
         UMU2 = UMUF*((FT*GT1 + GT*FT1)* TM2 +                            ANL63P
        *        (FT*GT2+ 2.*FT1*GT1 + GT*FT2)* TM1* TM1)                 ANL63P
         UMU3 = UMUF*((FT*GT1 + GT*FT1)* TM3 +                            ANL63P
        *        (FT*GT2+ 2.*FT1*GT1 + GT*FT2)* 3.* TM1* TM2 +            ANL63P
        *        (FT*GT3+ 3.*FT1*GT2 + 3.*FT2*GT1 + GT*FT3)*TM1**3)       ANL63P
         DOM(2) = UMU                                                     ANL63P
         DOM(3) = UMU1                                                    ANL63P
         DOM(4) = UMU2                                                    ANL63P
         DOM(5) = UMU3                                                    ANL63P
      50 RETURN                                                           ANL63P
         END                                                              ANL63P
```

SUBRØUTINE
APPLY

958

Purpose

APPLY computes the a vector defined by Equation 2.4-2 in Vol. I, together with the total first and second partials of a with respect to the state and the total first partials with respect to the costate, if needed.

```
      SUBROUTINE APPLY                                                  APPLY
C                                                                       APPLY
C                  THIS ROUTINE CONTROLS THE CALCULATION OF THE APPLIED APPLY
C                  ACCELERATIONS ON THE VEHICLE ,   THEIR TOTAL FIRST   APPLY
C                  AND SECOND PARTIALS WITH RESPECT TO THE STATE AND    APPLY
C                  THEIR TOTAL FIRST PARTIALS WITH RESPECT TO THE       APPLY
C                  COSTATE.  NOTE THAT THE TIME RATE OF CHANGE OF THE   APPLY
C                  MASS IS TREATED AS AN *APPLIED ACCELERATION*         APPLY
C                                                                       APPLY
      COMMON/GLOBAL/                                                    GLOBAL
     *GR     ,ER     ,OMGZ   ,XLAMRF,YMURF ,LUM    ,TO     ,EPSLON,INNER  GLOBAL
     *ITRMAX,JJOP(6)        ,IFATAL,NARC   ,NBRAN ,NFARC ,ID(4) ,KTAB(20),  GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,      GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                         GLOBAL
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
```

```
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2             MATS
      COMMON /MATS/                                                    MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24) MATS
      COMMON /MATS/                                                    MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,  MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,  MATS
     *POR   ,POO   ,PLG   ,PLP                                         MATS
      EQUIVALENCE(PROD1,PROD5)                                         MATS
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM  D
     *, LHT                                                            D
      COMMON /D/                                                       D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,  JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20) D
      DIMENSION NOM(20)                                                D
      EQUIVALENCE (NOM, V)                                             D
      COMMON /AXLE/                                                    AXLE
     *AV    ,AG    ,AP    ,AM    ,AVV   ,AGV   ,APV   ,AMV   ,AVG   ,  AXLE
     *AGG   ,APG   ,AMG   ,AVP   ,AGP   ,APP   ,AMP   ,AVR   ,AGR   ,  AXLE
     *APR   ,AMR   ,AVO   ,AGO   ,APO   ,AMO   ,AVU   ,AGU   ,APU   ,  AXLE
     *AMU   ,AVM   ,AGM   ,APM   ,AMM   ,AVZ   ,AGZ   ,APZ   ,AMZ   ,  AXLE
     *EAVV  ,EAGV  ,EAPV  ,EAMV  ,EAVG  ,EAGG  ,EAPG  ,EAMG  ,EAVP  ,  AXLE
     *EAGP  ,EAPP  ,EAMP  ,EAVR  ,EAGR  ,EAPR  ,EAMR  ,EAVO  ,EAGO  ,  AXLE
     *EAPO  ,EAMO  ,EAVU  ,EAGU  ,EAPU  ,EAMU  ,EAVM  ,EAGM  ,EAPM  ,  AXLE
     *EAMM  ,EAVZ  ,EAGZ  ,EAPZ  ,EAMZ  ,AVVV  ,AGVV  ,APVV  ,AMVV  ,  AXLE
     *AVGV  ,AGGV  ,APGV  ,AMGV  ,AVPV  ,AGPV  ,APPV  ,AMPV  ,AVRV  ,  AXLE
     *AGRV  ,APRV  ,AMRV  ,AVOV  ,AGOV  ,APOV  ,AMOV  ,AVUV  ,AGUV  ,  AXLE
     *APUV  ,AMUV  ,AVMV  ,AGMV  ,APMV  ,AMMV  ,AVZV  ,AGZV  ,APZV  ,  AXLE
     *AMZV  ,AVVG  ,AGVG  ,APVG  ,AMVG  ,AVGG  ,AGGG  ,APGG  ,AMGG  ,  AXLE
     *AVPG  ,AGPG  ,APPG  ,AMPG  ,AVRG  ,AGRG  ,APRG  ,AMRG  ,AVOG  ,  AXLE
     *AGOG  ,APOG  ,AMOG  ,AVUG  ,AGUG  ,APUG  ,AMUG  ,AVMG  ,AGMG  ,  AXLE
     *APMG  ,AMMG  ,AVZG  ,AGZG  ,APZG  ,AMZG  ,AVVP  ,AGVP  ,APVP  ,  AXLE
     *AMVP  ,AVGP  ,AGGP  ,APGP  ,AMGP  ,AVPP  ,AGPP  ,APPP  ,AMPP  ,  AXLE
     *AVRP  ,AGRP  ,APRP  ,AMRP  ,AVOP  ,AGOP  ,APOP  ,AMOP  ,AVUP  ,  AXLE
     *AGUP  ,APUP  ,AMUP  ,AVMP  ,AGMP  ,APMP  ,AMMP  ,AVZP  ,AGZP  ,  AXLE
     *APZP  ,AMZP  ,AVVR  ,AGVR  ,APVR  ,AMVR  ,AVGR  ,AGGR  ,APGR     AXLE
      COMMON /AXLE/                                                    AXLE
     *AMGR  ,AVPR  ,AGPR  ,APPR  ,AMPR  ,AVRR  ,AGRR  ,APRR  ,AMRR  ,  AXLE
     *AVOR  ,AGOR  ,APOR  ,AMOR  ,AVUR  ,AGUR  ,APUR  ,AMUR  ,AVMR  ,  AXLE
     *AGMR  ,APMR  ,AMMR  ,AVZR  ,AGZR  ,APZR  ,AMZR  ,AVVO  ,AGVO  ,  AXLE
     *APVO  ,AMVO  ,AVGO  ,AGGO  ,APGO  ,AMGO  ,AVPO  ,AGPO  ,APPO  ,  AXLE
     *AMPO  ,AVRO  ,AGRO  ,APRO  ,AMRO  ,AVOO  ,AGOO  ,APOO  ,AMOO  ,  AXLE
     *AVUO  ,AGUO  ,APUO  ,AMUO  ,AVMO  ,AGMO  ,APMO  ,AMMO  ,AVZO  ,  AXLE
     *AGZO  ,APZO  ,AMZO  ,AVVU  ,AGVU  ,APVU  ,AMVU  ,AVGU  ,AGGU  ,  AXLE
     *APGU  ,AMGU  ,AVPU  ,AGPU  ,APPU  ,AMPU  ,AVRU  ,AGRU  ,APRU  ,  AXLE
     *AMRU  ,AVOU  ,AGOU  ,APOU  ,AMOU  ,AVUU  ,AGUU  ,APUU  ,AMUU  ,  AXLE
     *AVMU  ,AGMU  ,APMU  ,AMMU  ,AVZU  ,AGZU  ,APZU  ,AMZU  ,AVVM  ,  AXLE
     *AGVM  ,APVM  ,AMVM  ,AVGM  ,AGGM  ,APGM  ,AMGM  ,AVPM  ,AGPM  ,  AXLE
     *APPM  ,AMPM  ,AVRM  ,AGRM  ,APRM  ,AMRM  ,AVOM  ,AGOM  ,APOM  ,  AXLE
     *AMOM  ,AVUM  ,AGUM  ,APUM  ,AMUM  ,AVMM  ,AGMM  ,APMM  ,AMMM  ,  AXLE
     *AVZM  ,AGZM  ,APZM  ,AMZM  ,AVVZ  ,AGVZ  ,APVZ  ,AMVZ  ,AVGZ  ,  AXLE
     *AGGZ  ,APGZ  ,AMGZ  ,AVPZ  ,AGPZ  ,APPZ  ,AMPZ  ,AVRZ  ,AGRZ  ,  AXLE
     *APRZ  ,AMRZ  ,AVOZ  ,AGOZ  ,APOZ  ,AMOZ  ,AVUZ  ,AGUZ  ,APUZ  ,  AXLE
     *AMUZ  ,AVMZ  ,AGMZ  ,APMZ  ,AMMZ  ,AVZZ  ,AGZZ  ,APZZ  ,AMZZ     AXLE
      COMMON /AXLE/                                                    AXLE
     *AVLV  ,AGLV  ,APLV  ,AMLV  ,AVLG  ,AGLG  ,APLG  ,AMLG  ,AVLP  ,  AXLE
     *AGLP  ,APLP  ,AMLP  ,AVVLV ,AGVLV ,APVLV ,AMVLV ,AVGLV ,AGGLV ,  AXLE
     *APGLV ,AMGLV ,AVPLV ,AGPLV ,APPLV ,AMPLV ,AVRLV ,AGRLV ,APRLV ,  AXLE
     *AMRLV ,AVOLV ,AGOLV ,APOLV ,AMOLV ,AVULV ,AGULV ,APULV ,AMULV ,  AXLE
     *AVMLV ,AGMLV ,APMLV ,AMMLV ,AVZLV ,AGZLV ,APZLV ,AMZLV ,AVVLG ,  AXLE
     *AGVLG ,APVLG ,AMVLG ,AVGLG ,AGGLG ,APGLG ,AMGLG ,AVPLG ,AGPLG ,  AXLE
     *APPLG ,AMPLG ,AVRLG ,AGRLG ,APRLG ,AMRLG ,AVOLG ,AGOLG ,APOLG ,  AXLE
     *AMOLG ,AVULG ,AGULG ,APULG ,AMULG ,AVMLG ,AGMLG ,APMLG ,AMMLG ,  AXLE
     *AVZLG ,AGZLG ,APZLG ,AMZLG ,AVVLP ,AGVLP ,APVLP ,AMVLP ,AVGLP ,  AXLE
     *AGGLP ,APGLP ,AMGLP ,AVPLP ,AGPLP ,APPLP ,AMPLP ,AVRLP ,AGRLP ,  AXLE
     *APRLP ,AMRLP ,AVOLP ,AGOLP ,APOLP ,AMOLP ,AVULP ,AGULP ,APULP ,  AXLE
     *AMULP ,AVMLP ,AGMLP ,APMLP ,AMMLP ,AVZLP ,AGZLP ,APZLP ,AMZLP     AXLE
      COMMON /CNTRL/                                                   CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,  CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,  CNTRL
```

958

```
      *KPAGE ,NNP    ,NUP    ,IARC   ,TRSTR  ,IMAX   ,KTIME  ,KONVER,NOPRNT,      CNTRL
      *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES                   CNTRL
       LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                             CNTRL
       DIMENSION AY(1), AYIMP(32), AW(12), AXLE(1)                                APPLY
       DIMENSION AWY(4, 24), WW(4, 8), AYY(4, 1), ZZ(4, 3), AWW(4, 9),            APPLY
      *AYLY(4, 1), AWLY(4, 9), AWG(4, 3)                                          APPLY
       EQUIVALENCE (AY, AVV), (AVT, AW(1)), (AVD, AW(5)), (AVA, AW(9)),           APPLY
      *(AGT, AW(2)), (AGD, AW(6)), (AGA, AW(10)), (APT, AW(3)),                   APPLY
      *(APD, AW(7)), (APA, AW(11)), (AMT, AW(4)), (AXLE, AV)                      APPLY
       EQUIVALENCE (AYY, AVVV), (AYLY, AVVLV), (AVTM, AWY(1, 19)),                APPLY
      *(AVDM, AWY(1, 20)), (AVAM, AWY(1, 21)), (AGTG, AWG(2, 1)),                 APPLY
      *(AGDG, AWG(2, 2)), (AGAG, AWG(2, 3)), (AGAV, AWY(2, 3)),                   APPLY
      *(AGAR, AWY(2, 12)), (AGTM, AWY(2, 19)), (AGDM, AWY(2, 20)),                APPLY
      *(AGAM, AWY(2, 21)), (APTG, AWG(3, 1)), (APDG, AWG(3, 2)),                  APPLY
      *(APAG, AWG(3, 3)), (APAV, AWY(3, 3)), (APAR, AWY(3, 12)),                  APPLY
      *(APTM, AWY(3, 19)), (APDM, AWY(3, 20)), (APAM, AWY(3, 21)),                APPLY
      *(AMTR, AWY(4, 10)), (AVTD, AWW(1, 4)), (AVTA, AWW(1, 7)),                  APPLY
      *(AGTD, AWW(2, 4)), (AGTA, AWW(2, 7)), (APTD, AWW(3, 4)),                   APPLY
      *(APTA, AWW(3, 7)), (AMTT, AWW(4, 1)), (AVDT, AWW(1, 2)),                   APPLY
      *(AVDD, AWW(1, 5)), (AVDA, AWW(1, 8)), (AGDT, AWW(2, 2)),                   APPLY
      *(AGDD, AWW(2, 5)), (AGDA, AWW(2, 8)), (APDT, AWW(3, 2)),                   APPLY
      *(APDD, AWW(3, 5)), (APDA, AWW(3, 8)), (AVAT, AWW(1, 3)),                   APPLY
      *(AVAD, AWW(1, 6)), (AVAA, AWW(1, 9)), (AGAT, AWW(2, 3)),                   APPLY
      *(AGAD, AWW(2, 6)), (AGAA, AWW(2, 9)), (APA T, AWW(3, 3)),                  APPLY
      *(APAD, AWW(3, 6)), (APAA, AWW(3, 9)), (AGTLG, AWLY(2, 4)),                 APPLY
      *(AGDLG, AWLY(2, 5)), (AGALG, AWLY(2, 6)), (APTLG, AWLY(3, 4)),             APPLY
      *(APDLG, AWLY(3, 5)), (APALG, AWLY(3, 6)), (AGTLP, AWLY(2, 7)),             APPLY
      *(AGDLP, AWLY(2, 8)), (AGALP, AWLY(2, 9)), (APTLP, AWLY(3, 7)),             APPLY
      *(APDLP, AWLY(3, 8)), (APALP, AWLY(3, 9))                                   APPLY
       EQUIVALENCE (AMTV, AWY(4, 1))                                              JUL21
       EQUIVALENCE(AVAV, AWY(1, 3)), (AVAR, AWY(1, 12))                           APPLY
       EQUIVALENCE (AYIMP, WW)                                                    APPLY
       DATA AW/12*0./, AWY/96*0./, AWW/36*0./, AWLY/36*0./, AWG/12*0./            APPLY
       DO 40 I = 1, 80                                                            APPLY
    40 AXLE(I) = 0.                                                               APPLY
       TCDAE = T*CODAE                                                            APPLY
       TSDAE = T*SIDAE                                                            APPLY
       DBCA = DB*COSA                                                             APPLY
       DBSA = DB*SINA                                                             APPLY
       DBRCA = DBR*COSA                                                           APPLY
       DBRSA = DBR*SINA                                                           APPLY
       GEISP = GR*ISP                                                             APPLY
       CONST = 3600.*GR                                                           JUL21
       AV = (TCDAE - DBCA - DRAG)/M                                               APPLY
       AM = -FVAC/GEISP                                                           APPLY
       IF(J1 .EQ. 4) AM = -SFC*T/CONST                                            JUL21
       IF(KODE .EQ. 4) GO TO 51                                                   APPLY   51
       AX = (TSDAE - DBSA + LIFT)/M                                               APPLY
       AG = AX*COSPHI                                                             APPLY
       AP = AX*SINPHI                                                             APPLY
C      COMPUTE THE EXPLICIT FIRST PARTIALS OF THE A-VECTOR WITH RESPECT           APPLY
C      TO THE STATE AND DECISION VECTORS.                                         APPLY
    51 AVV = -DRAGV/M                                                             APPLY
       AVR = (-DBRCA - DRAGR)/M                                                   APPLY
       AVM = -AV/M                                                                APPLY
       IF(J1 - 4) 511, 512, 512                                                   JUL21   511 512
   511 AMT = -(1. - ISPF*FVAC/ISP)/GEISP                                          JUL21
       AMV = 0.                                                                   JUL21
       AMR = AE*PAR*AMT                                                           APPLY
       GO TO 513                                                                  JUL21   513
   512 AMT = -SFC/CONST                                                           JUL21
       AMV = -SFCV*T/CONST                                                        JUL21
       AMR = -SFCR*T/CONST                                                        JUL21
   513 AVT = CODAE/M                                                              JUL21
       AVD = TSDAE/M                                                              APPLY
       AVA = (-TSDAE + DBSA - DRAGA)/M                                            APPLY
       AXT = SIDAE/M                                                              APPLY
       AXD = -TCDAE/M                                                             APPLY
       IF(KODE .EQ. 4) GO TO 52                                                   APPLY   52
       AXV = LIFTV/M                                                              APPLY
       AXR = (-DBRSA + LIFTR)/M                                                   APPLY
       AXM = (LIFTM - AX)/M                                                       APPLY
       AGV = AXV*COSPHI                                                           APPLY
```

```
         AGR = AXR*COSPHI                                                  APPLY
         AGM = AXM*COSPHI                                                  APPLY
         APV = AXV*SINPHI                                                  APPLY
         APR = AXR*SINPHI                                                  APPLY
         APM = AXM*SINPHI                                                  APPLY
         AXA = (TCDAE - DBCA + LIFTA)/M                                    APPLY
         AGT = AXT*COSPHI                                                  APPLY
         AGD = AXD*COSPHI                                                  APPLY
         AGA = AXA*COSPHI                                                  APPLY
         APT = AXT*SINPHI                                                  APPLY
         APD = AXD*SINPHI                                                  APPLY
         APA = AXA*SINPHI                                                  APPLY
C        IS ALPHA NONOPTIMAL.                                              APPLY
      52 IF(KODE .GT. 2) GO TO 102                                         APPLY
C        COMPUTE THE STARRED FIRST PARTIALS OF THE A-VECTOR WITH RESPECT   APPLY
C        TO THE STATE.                                                     APPLY
         CALL MATMLT(AVIMP, AVT, DEPDEV, 4, 2, 8)                          APPLY
         CALL MATADD(EAVV, AVV, AVIMP, 4, 8)                               APPLY
C        COMPUTE THE TOTAL FIRST PARTIALS OF THE A-VECTOR WITH RESPECT TO  APPLY
C        THE COSTATE.                                                      APPLY
     101 IF(KONVER) RETURN                                                 APPLY
         CALL MATMLT(AVLV, AVT, DPDL, 4, 3, 3)                             APPLY
C        ADD THE CONTRIBUTIONS DUE TO PHI TO THE ABOVE PARTIALS.           APPLY
         AGLG = AGLG - AP*PLG                                              APPLY
         AGLP = AGLP - AP*PLP                                              APPLY
         APLG = APLG + AG*PLG                                              APPLY
         APLP = APLP + AG*PLP                                              APPLY
         GO TO 103                                                         APPLY
C        COMPUTE THE STARRED FIRST PARTIALS OF THE A-VECTOR WITH RESPECT   APPLY
C        TO THE STATE                                                      APPLY
     102 CALL MATMLT(AVIMP, AVT, DPDY, 4, 3, 8)                            APPLY
C        IS PHI NONOPTIMAL                                                 APPLY
         IF(KODE .GT. 3) GO TO 104                                         APPLY
         CALL MATADD(EAVV, AVV, AVIMP, 4, 8)                               APPLY
         GO TO 101                                                         APPLY
C        COMPUTE THE CONTRIBUTIONS DUE TO THE DECISION VECTOR TO THE TOTAL APPLY
C        FIRST PARTIALS OF THE A-VECTOR WITH RESPECT TO THE STATE.         APPLY
     103 CALL MATMLT(AVIMP, AVT, DPDY, 4, 3, 8)                            APPLY
C        COMPUTE THE TOTAL FIRST PARTIALS OF THE A-VECTOR WITH RESPECT TO  APPLY
C        THE STATE.                                                        APPLY
     104 CALL MATADD(AVV, AVV, AVIMP, 4, 8)                                APPLY
C        ADD THE CONTRIBUTIONS DUE TO PHI TO THE ABOVE PARTIALS.           APPLY
         AGG = AGG - AP*PG                                                 APPLY
         APG = APG + AG*PG                                                 APPLY
C        IS THIS A CONVERGED TRAJECTORY.                                   APPLY
         IF(KONVER) RETURN                                                 APPLY
C        CLEAR THE ARRAY OF SECOND PARTIALS WITH RESPECT TO THE STATE      APPLY
         DO 105 I = 81, 324                                                APPLY
     105 AXLE(I)= 0.                                                       APPLY
C        COMPUTE THE EXPLICIT SECOND PARTIALS WITH RESPECT TO THE STATE.   APPLY
         IF(J1 - 4) 10501, 10502, 10502                                    JUL21
   10501 AMTT = (ISPFF*FVAC + 2.*ISPF*(1. - ISPF*FVAC/ISP))/GEISP/ISP      JUL21
         AMTV = 0.                                                         JUL21
         AMTR = AE*PAR*AMTT                                                APPLY
         GO TO 10503                                                       JUL21
   10502 AMTT = 0.                                                         JUL21
         AMTV = -SFCV/CONST                                                JUL21
         AMTR = -SFCH/CONST                                                JUL21
   10503 AVVV = -DRAGVV/M                                                  JUL21
         AVRV = -DRAGVR/M                                                  APPLY
         AVMV = DRAGV/M**2                                                 APPLY
         AVVR = AVRV                                                       APPLY
         AVRR = -(DBRR*COSA + DRAGRR)/M                                    APPLY
         AVMR = (DBRCA + DRAGR)/M**2                                       APPLY
         AVVM = AVMV                                                       APPLY
         AVRM = AVMR                                                       APPLY
         AVMM = 2.*AV/M**2                                                 APPLY
         IF(J1 - 4) 10504, 10505, 10505                                    JUL21
   10504 AMVR = 0.                                                         JUL21
         AMRV = 0.                                                         JUL21
         AMVV = 0.                                                         JUL21
```

```
      AMRR = AE*(PARR*AMT + PAR*AMTR)                              APPLY
      GO TO 10506                                                  JUL21   10506
10505 AMVV = -SFCVV*T/CONST                                        JUL21
      AMVR = -SFCVH*T/CONST                                        JUL21
      AMRV = AMVR                                                  JUL21
      AMRR = -SFCHH*T/CONST                                        JUL21
10506 IF(KODE .EQ. 4) GO TO 1051                                   JUL21   1051
      AXVY = LIFTVV/M                                              APPLY
      AXRY = LIFTVR/M                                              APPLY
      AXMY = (LIFTVM - AXV)/M                                      APPLY
      AGVV = AXVY*COSPHI                                           APPLY
      AGRV = AXRY*COSPHI                                           APPLY
      AGMV = AXMY*COSPHI                                           APPLY
      APVV = AXVY*SINPHI                                           APPLY
      APRV = AXRY*SINPHI                                           APPLY
      APMV = AXMY*SINPHI                                           APPLY
      TEMP1= SINPHI*PG                                             APPLY
      AGVG = -AXV*TEMP1                                            APPLY
      AGRG = -AXR*TEMP1                                            APPLY
      AGMG = -AXM*TEMP1                                            APPLY
      TEMP2= COSPHI*PG                                             APPLY
      APVG = AXV*TEMP2                                             APPLY
      APRG = AXR*TEMP2                                             APPLY
      APMG = AXM*TEMP2                                             APPLY
      AXRY = (LIFTRR - DBRR*SINA)/M                                APPLY
      AXMY = (LIFTRM - AXR)/M                                      APPLY
      AGVR = AGRV                                                  APPLY
      AGRR = AXRY*COSPHI                                           APPLY
      AGMR = AXMY*COSPHI                                           APPLY
      APVR = APRV                                                  APPLY
      APRR = AXRY*SINPHI                                           APPLY
      APMR = AXMY*SINPHI                                           APPLY
      AXMY = (LIFTMM - 2.*AXM)/M                                   APPLY
      AGVM = AGMV                                                  APPLY
      AGRM = AGMR                                                  APPLY
      AGMM = AXMY*COSPHI                                           APPLY
      APVM = APMV                                                  APPLY
      APRM = APMR                                                  APPLY
      APMM = AXMY*SINPHI                                           APPLY
C     COMPUTE THE EXPLICIT SECOND PARTIALS WITH RESPECT TO THE STATE APPLY
C     AND DECISION VECTORS                                         APPLY
 1051 AVAV = -DRAGVA/M                                             APPLY
      AVAR = (DBRSA - DRAGRA)/M                                    APPLY
      AVTM = -AVT/M                                                APPLY
      AVDM = -AVD/M                                                APPLY
      AVAM = -AVA/M                                                APPLY
      IF(KODE .EQ. 4) GO TO 1052                                   APPLY   1052
      AXVA = LIFTVA/M                                              APPLY
      AXRA = (LIFTRA - DBRCA)/M                                    APPLY
      AXMT = -AXT/M                                                APPLY
      AXMD = -AXD/M                                                APPLY
      AXMA = (LIFTMA - AXA)/M                                      APPLY
      AGAV = AXVA*COSPHI                                           APPLY
      AGAR = AXRA*COSPHI                                           APPLY
      AGTM = AXMT*COSPHI                                           APPLY
      AGDM = AXMD*COSPHI                                           APPLY
      AGAM = AXMA*COSPHI                                           APPLY
      APAV = AXVA*SINPHI                                           APPLY
      APAR = AXRA*SINPHI                                           APPLY
      APTM = AXMT*SINPHI                                           APPLY
      APDM = AXMD*SINPHI                                           APPLY
      APAM = AXMA*SINPHI                                           APPLY
C     COMPUTE THE EXPLICIT SECOND PARTIALS WITH RESPECT TO THE DECISION APPLY
C     VECTOR.                                                      APPLY
 1052 AVTD = AXT                                                   APPLY
      AVTA =-AXT                                                   APPLY
      AVDT = AVTD                                                  APPLY
      AVDD = AXD                                                   APPLY
      AVDA =-AXD                                                   APPLY
      AVAT = AVTA                                                  APPLY
      AVAD = AVDA                                                  APPLY
      AVAA = (-TCDAE +DBCA - DRAGAA)/M                             APPLY
      IF(KODE .EQ. 4) GO TO 1053                                   APPLY   1053
```

```
      AXTD =-AVT                                                        APPLY
      AXTA = AVT                                                        APPLY
      AGTD = AXTD*COSPHI                                                APPLY
      AGTA = AXTA*COSPHI                                                APPLY
      APTD = AXTD*SINPHI                                                APPLY
      APTA = AXTA*SINPHI                                                APPLY
      AXDD =-AVD                                                        APPLY
      AXDA = AVD                                                        APPLY
      AGDT = AGTD                                                       APPLY
      AGDD = AXDD*COSPHI                                                APPLY
      AGDA = AXDA*COSPHI                                                APPLY
      APDT = APTD                                                       APPLY
      APDD = AXDD*SINPHI                                                APPLY
      APDA = AXDA*SINPHI                                                APPLY
      AXAA = (-TSDAE + DBSA + LIFTAA)/M                                 APPLY
      AGAT = AGTA                                                       APPLY
      AGAD = AGDA                                                       APPLY
      AGAA = AXAA*COSPHI                                                APPLY
      APAT = APTA                                                       APPLY
      APAD = APDA                                                       APPLY
      APAA = AXAA*SINPHI                                                APPLY
 1053 KSTAR = 2                                                         APPLY
      IF(KODE .GT. 2) KSTAR = 3                                         APPLY
      II = -7                                                           APPLY
      III = -2                                                          APPLY
      DO 109 I = 1, 8                                                   APPLY
      II = II + 8                                                       APPLY
      III = III + 3                                                     APPLY
      JJ = -2                                                           APPLY
      DO 106 J = 1, 8                                                   APPLY
      JJ = JJ + 3                                                       APPLY
  106 CALL MATMLT(WW(1, J), AWY(1, JJ), DPDY(1, I), 4, 3, 1)            APPLY
      CALL MATADD(AYY(1, II), AYY(1, II), WW, 4, 8)                     APPLY
      KK = -2                                                           APPLY
      DO 107 K = 1, KSTAR                                               APPLY
      KK = KK + 3                                                       APPLY
  107 CALL MATMLT(ZZ(1, K), AWW(1, KK), DPDY(1, I), 4, 3, 1)            APPLY
      CALL MATADD(ZZ, ZZ, AWY(1, III), 4, KSTAR)                        APPLY
      IF(KODE .GT. 2) GO TO 108                                         APPLY  108
      CALL MATMLT(WW, ZZ, DEPDEY, 4, 2, 8)                              APPLY
      CALL MATADD(AYY(1, II), AYY(1, II), WW, 4, 8)                     APPLY
      CALL MATMLT(WW, AVT, PROD1(1, II), 4, 2, 8)                       APPLY
      GO TO 109                                                         APPLY  109
  108 CALL MATMLT(WW, ZZ, DPDY, 4, 3, 8)                                APPLY
      CALL MATADD(AYY(1, II), AYY(1, II), WW, 4, 8)                     APPLY
      CALL MATMLT(WW, AVT, PROD5(1, II), 4, 3, 8)                       APPLY
  109 CALL MATADD(AYY(1, II), AYY(1, II), WW, 4, 8)                     APPLY
      IF(KODE .GT. 3) RETURN                                            APPLY
      AGTG = -AXT*TEMP1                                                 APPLY
      AGDG = -AXD*TEMP1                                                 APPLY
      AGAG = -AXA*TEMP1                                                 APPLY
      APTG =  AXT*TEMP2                                                 APPLY
      APDG =  AXD*TEMP2                                                 APPLY
      APAG=  AXA*TEMP2                                                  APPLY
      IF(KODE .EQ. 3) GO TO 1091                                        APPLY  1091
      CALL MATMLT(WW, AWG, DEPDEY, 4, 2, 8)                             APPLY
      GO TO 1092                                                        APPLY  1092
 1091 CALL MATMLT(WW, AWG, DPDY, 4, 3, 8)                               APPLY
 1092 CALL MATADD(AYY(1, 9), AYY(1, 9), WW, 4, 8)                       APPLY
C     CLEAR THE ARRAY OF SECOND PARTIALS WITH RESPECT TO THE STATE AND  APPLY
C     COSTATE.                                                          APPLY
      DO 110 I = 337, 432                                               APPLY
  110 AXLE (I)= 0.                                                      APPLY
      IF(KODE .EQ. 2) GO TO 1101                                        APPLY  1101
C     COMPUTE THE EXPLICIT SECOND PARTIALS WITH RESPECT TO THE STATE AND APPLY
C     COSTATE.                                                          APPLY
      TEMP1 = SINPHI*PLG                                                APPLY
      TEMP2 = COSPHI*PLG                                                APPLY
      TEMP3 = SINPHI*PLP                                                APPLY
      TEMP4 = COSPHI*PLP                                                APPLY
      AGVLG =-AXV*TEMP1                                                 APPLY
      AGRLG =-AXR*TEMP1                                                 APPLY
      AGMLG =-AXM*TEMP1                                                 APPLY
```

962

```
      APVLG = AXV*TEMP2                                    APPLY
      APRLG = AXR*TEMP2                                    APPLY
      APMLG = AXM*TEMP2                                    APPLY
      AGVLP =-AXV*TEMP3                                    APPLY
      AGRLP =-AXR*TEMP3                                    APPLY
      AGMLP =-AXM*TEMP3                                    APPLY
      APVLP = AXV*TEMP4                                    APPLY
      APRLP = AXR*TEMP4                                    APPLY
      APMLP = AXM*TEMP4                                    APPLY
      AGTLG =-AXT*TEMP1                                    APPLY
      AGDLG =-AXD*TEMP1                                    APPLY
      AGALG =-AXA*TEMP1                                    APPLY
      APTLG = AXT*TEMP2                                    APPLY
      APDLG = AXD*TEMP2                                    APPLY
      APALG = AXA*TEMP2                                    APPLY
      AGTLP =-AXT*TEMP3                                    APPLY
      AGDLP =-AXD*TEMP3                                    APPLY
      AGALP =-AXA*TEMP3                                    APPLY
      APTLP = AXT*TEMP4                                    APPLY
      APDLP = AXD*TEMP4                                    APPLY
      APALP = AXA*TEMP4                                    APPLY
 1101 III= -2                                              APPLY
      II = -7                                              APPLY
      DO 114 I = 1, 3                                      APPLY
      II = II + 8                                          APPLY
      III = III +3                                         APPLY
      IF(KODE .EQ. 3) GO TO 113                            APPLY  113
      JJ = -2                                              APPLY
      DO 111 J = 1, 8                                      APPLY
      JJ = JJ + 3                                          APPLY
  111  CALL MATMLT(WW(1, J), AWY(1, JJ), DPDL(1, I), 4, 3, 1)   APPLY
      CALL MATADD(AYLY(1, II), AYLY(1, II), WW, 4, 8)      APPLY
      KK = -2                                              APPLY
      DO 112 K = 1, 2                                      APPLY
      KK = KK + 3                                          APPLY
  112 CALL MATMLT(ZZ(1, K), AWW(1, KK), DPDL(1, I), 4, 3, 1)   APPLY
      CALL MATADD(ZZ, ZZ, AWLY(1, III), 4, 2)              APPLY
      CALL MATMLT(WW, ZZ, DEPDEY, 4, 2, 8)                 APPLY
      CALL MATADD(AYLY(1, II), AYLY(1, II), WW, 4, 8)      APPLY
      CALL MATMLT(WW, AVT, PROD9(1, II), 4, 2, 8)          APPLY
      GO TO 114                                            APPLY  114
  113 CALL MATMLT(WW, AWLY(1, III), DPDY, 4, 3, 8)         APPLY
  114 CALL MATADD(AYLY(1, II), AYLY(1, II), WW, 4, 8)      APPLY
      RETURN                                               APPLY
      END                                                  APPLY
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AE	A_{exit}	I	Total nozzle exit area (FT^2)	/DYNA	89	APPLY ARCIN IMPULS NLDRV TH2	I O I I I	AE AE AE AE AE
APLP	$\partial a^v/\partial\lambda_v$	M	The first entry in a 4x3 matrix that contains $\partial a/\partial\lambda_v$, $\partial a/\partial\lambda_y$ and $\partial a/\partial\lambda_\phi$	/AXLE	335	APPLY NLDRV	M I	APLP APLP
AV	a^v	M	The first word of a four word array that contains the vector of applied accelerations defined by Equation 2.4-2 in Vol.I of this document.	/AXLE	1	APPLY APPLY NLDRV	M O I	AV AXLE AV
AVV	$\partial a^v/\partial v$	M	The first entry in a 4x8 matrix that contains the total partial derivatives of the a vector with respect to the QL state vector (excluding the heating state), $\partial a/\partial y$.	/AXLE	5	APPLY NLDRV	M I	AVV AVV
AVVR	$\partial(\delta a^v/\delta V)/\partial h$	O	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial h$	/AXLE	165	APPLY NLDRV	O I	AVVR AVVR
AVVV	$\partial(\delta a^v/\delta V)/\partial V$	O	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial V$.	/AXLE	69	APPLY APPLY NLDRV	O I I	AVVV AYY AVVV
AW		D	A 4x3 array containing A_w	/APPLY	*	APPLY	D	AW
AWG		C	A 4x3 array containing $A_{w\phi}\phi_y$	/APPLY	*	APPLY	C	AWG
AWLY		C	A 4x9 array containing $\partial(A_w)/\partial\lambda_v$, $\partial(A_w)/\partial\lambda_y$ and $\partial(A_w)/\partial\lambda_\phi$	/APPLY	*	APPLY	C	AWLY
AWW		C	A 4x9 array containing A_{ww}	/APPLY	*	APPLY	C	AWW
AWY		C	A 4x24 array containing A_{wy}	/APPLY	*	APPLY	C	AWY
AXLE	a^v	O	The first word of a four word array that contains the vector of applied accelerations defined by Equation 2.4-2 in Vol.I of this document.	/AXLE	1	APPLY APPLY NLDRV	M O I	AV AXLE AV
AYY	$\partial(\delta a^v/\delta V)/\partial V$	I	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial V$.	/AXLE	69	APPLY APPLY NLDRV	O I I	AVVV AYY AVVV
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol	/DYNA	151	AL1 AL4 AL6 AL7 AL8 AL9 APPLY CONTRL NLDRV TH3 UT	I I I I I I I I I I O	CODAE CODAE CODAE CODAE CODAE CODAE CODAE CODAE CODAE CODAE CODAE
COSA	$\cos\alpha$	I	See symbol	/DYNA	10	AL1 AL4 AL6 AL7 AL8 AL9 APPLY CONTRL NLDRV OUTPUT TH3 UT	I I I I I I I I I I I M	COSA COSA COSA COSA COSA COSA COSA COSA COSA COSA COSA COSA
COSPHI	$\cos\phi$	I	See symbol	/DYNA	93	AL1 AL4 APPLY ARCIN CONTRL OUTPUT	I I I O M I	COSPHI COSPHI COSPHI COSPHI COSPHI COSPHI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DB	D_b	I	Base drag (LBS)	/DYNA	/(163)	AL1	I	DB
						AL4	I	DB
						AL6	I	DB
						AL7	I	DB
						AL8	I	DB
						AL9	I	DB
						APPLY	I	DB
						CONTRL	I	DB
						NLDRV	I	DB
						OUTPUT	I	DB
						STATEF	I	DB
						TH3	I	DB
						UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol	/DYNA	/(86)	AL1	I	DBR
						AL4	I	DBR
						AL6	I	DBR
						AL7	I	DBR
						AL8	I	DBR
						AL9	I	DBR
						APPLY	I	DBR
						STATEF	I	DBR
						TH3	I	DBR
						UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol	/DYNA	/(87)	AL4	I	DBRR
						AL6	I	DBRR
						AL7	I	DBRR
						AL8	I	DBRR
						AL9	I	DBRR
						APPLY	I	DBRR
						STATEF	I	DBRR
						TH3	I	DBRR
						UT	I	DBRR
DEPDEY	$\delta p/\delta y$	I	A 2x8 matrix that contains $\delta p/\delta y = \partial p/\partial y\|_m$ = constant	/MATS	/(285)	ALGCON	M	DEPDEY
						APPLY	I	DEPDEY
DPDL	$\partial m/\partial\lambda$	I	A 3x3 matrix that contains $\partial m/\partial\lambda_v$, $\partial m/\partial\lambda_\gamma$ and $\partial m/\partial\lambda_\phi$	/MATS	/(301)	ALGCON	I	DPDL
						APPLY	I	DPDL
DPDY	$\partial m/\partial y$	I	A 3x8 matrix that contains the total first partial derivatives of the in-plane-control vector wrt the BL state.	/MATS	/(261)	ALGCON	I	DPDY
						APPLY	I	DPDY
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA	/(69)	AL5	I	DRAG
						AL7	I	DRAG
						AL8	I	DRAG
						AL9	I	DRAG
						APPLY	I	DRAG
						CONTRL	I	DRAG
						ENVPRO	I	DRAG
						NLDRV	I	DRAG
						OUTPUT	I	DRAG
						TH3	I	DRAG
						UT	M	DRAG
DRAGA	$\partial D/\partial\alpha$	I	See symbol	/DYNA	/(72)	AL1	I	DRAGA
						AL5	I	DRAGA
						AL7	I	DRAGA
						AL8	I	DRAGA
						AL9	I	DRAGA
						APPLY	I	DRAGA
						TH3	I	DRAGA
						UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial\alpha^2$	I	See symbol	/DYNA	/(78)	AL1	I	DRAGAA
						AL5	I	DRAGAA
						AL7	I	DRAGAA
						AL8	I	DRAGAA
						AL9	I	DRAGAA
						APPLY	I	DRAGAA
						TH3	I	DRAGAA
						UT	M	DRAGAA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGR	$\partial D/\partial R$	I	See symbol	/DYNA	/(71)	AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I M	DRAGR DRAGR DRAGR DRAGR DRAGR DRAGR DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	I	See symbol	/DYNA	/(77)	AL1 AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I I M	DRAGRA DRAGRA DRAGRA DRAGRA DRAGRA DRAGRA DRAGRA DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol	/DYNA	/(76)	AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I M	DRAGRR DRAGRR DRAGRR DRAGRR DRAGRR DRAGRR DRAGRR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA	/(70)	AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I M	DRAGV DRAGV DRAGV DRAGV DRAGV DRAGV DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	I	See symbol	/DYNA	/(75)	AL1 AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I I M	DRAGVA DRAGVA DRAGVA DRAGVA DRAGVA DRAGVA DRAGVA DRAGVA
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA	/(74)	AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I M	DRAGVR DRAGVR DRAGVR DRAGVR DRAGVR DRAGVR DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA	/(73)	AL5 AL7 AL8 AL9 APPLY TH3 UT	I I I I I I M	DRAGVV DRAGVV DRAGVV DRAGVV DRAGVV DRAGVV DRAGVV
EAVV	$\delta a^v/\delta v$	I	The first entry in a 4x8 matrix that contains, when a is optimal, $\delta a/\delta y = \partial a/\partial y\|_a$ = constant When a is nonoptimal, $\delta a/\delta y = \partial a/\partial y$	/AXLE	/(37)	APPLY NLDRV	I I	EAVV EAVV
FVAC		I	Total vacuum thrust (rocket) (LBS)	/DYNA	/(33)	APPLY ARCIN IMPULS NLDRV STATEF TH2	I M M I M I	FVAC FVAC FVAC FVAC FVAC FVAC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCOL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
ISP	I_{SP}	I	Vacuum specific impulse (SECS)	/DYNA /(	45)	APPLY	I	ISP
						ARCIN	O	ISP
						IMPULS	O	ISP
ISPF	$\partial I_{SP}/\partial T$	I	See symbol	/DYNA /(	179)	APPLY	I	ISPF
						IMPULS	O	ISPF
ISPFF	$\partial^2 I_{SP}/\partial T^2$	I	See symbol	/DYNA /(	180)	APPLY	I	ISPFF
						IMPULS	O	ISPFF
J1		I	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY	I	J1
						ARCIN	O	J1
						CONTRL	M	J1
						FORCES	I	J1
						NPLANE	I	J1
						STATEF	I	J1
						THROTL	M	J1
KODE		I	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.	/DYNA /(	25)	APPLY	I	KODE
						ARCIN	O	KODE
						CONTRL	M	KODE
						FORCES	I	KODE
						NLDRV	I	KODE
						STATEF	I	KODE
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON	I	KONVER
						APPLY	I	KONVER
						ARCIN	I	KONVER
						COMOMO	O	KONVER
						GROPE	O	KONVER
						NLDRV	I	KONVER
						OUTPUT	I	KONVER
						RKUTT1	I	KONVER
LIFT	L	I	Aerodynamic lift (LBS)	/DYNA /(	60)	AL4	I	LIFT
						AL5	I	LIFT
						AL6	I	LIFT
						APPLY	I	LIFT
						CONTRL	I	LIFT
						ENVPRQ	I	LIFT
						OUTPUT	I	LIFT
						TH3	I	LIFT
						UT	O	LIFT
LIFTA	$\partial L/\partial \alpha$	I	See symbol	/DYNA /(	63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTAA	$\partial^2 L/\partial \alpha^2$	I	See symbol	/DYNA /(	144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA

967

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTM	$\partial L/\partial m$	I	See symbol	/DYNA /(	81)	AL4	I	LIFTM
						AL5	I	LIFTM
						AL6	I	LIFTM
						APPLY	I	LIFTM
						TH3	I	LIFTM
						UT	O	LIFTM
LIFTMA	$\partial^2 L/\partial m \partial \alpha$	I	See symbol	/DYNA /(	85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA
LIFTMM	$\partial^2 L/\partial m^2$	I	See symbol	/DYNA /(	84)	AL4	I	LIFTMM
						AL5	I	LIFTMM
						AL6	I	LIFTMM
						APPLY	I	LIFTMM
						TH3	I	LIFTMM
						UT	O	LIFTMM
LIFTR	$\partial L/\partial R$	I	See symbol	/DYNA /(	62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	I	See symbol	/DYNA /(	68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R \partial m$	I	See symbol	/DYNA /(	83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	I	See symbol	/DYNA /(	67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	I	See symbol	/DYNA /(	61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	I	See symbol	/DYNA /(	66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTVM	$\partial^2 L/\partial V \partial m$	I	See symbol	/DYNA /(	82)	AL4	I	LIFTVM
						AL5	I	LIFTVM
						AL6	I	LIFTVM
						APPLY	I	LIFTVM
						TH3	I	LIFTVM
						UT	O	LIFTVM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LIFTVR	$\partial^2 L/\partial V \partial R$	I	See symbol	/DYNA	/(65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	I	See symbol	/DYNA	/(64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV
M	m	I	Mass (G'S)	/D	/(97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
PAR	$\partial P_s/\partial R$	I	See symbol	/DYNA	/(18)	APPLY	I	PAR
						TH2	I	PAR
PARR	$\partial^2 P_s/\partial R^2$	I	See symbol	/DYNA	/(22)	APPLY	I	PARR
						TH2	I	PARR
PG	ϕ_γ	I	See symbol	/MATS	/(551)	AL4	M	PG
						APPLY	I	PG
						ARCIN	O	PG
						CONTRL	O	PG
PLG	ϕ_{λ_γ}	I	See symbol	/MATS	/(570)	APPLY	I	PLG
						ARCIN	O	PLG
						CONTRL	O	PLG
PLP	ϕ_{λ_ψ}	I	See symbol	/MATS	/(571)	APPLY	I	PLP
						ARCIN	O	PLP
						CONTRL	O	PLP
PROD1	$\partial(\delta p/\delta y)/\partial y$	I	A 2x64 matrix that contains the total first partials of the matrix DEPDEY wrt the OL state.	/MATS	/(310)	ALGCON	I	PROD1
						ALGCON	I	PROD5
						APPLY	I	PROD1
						APPLY	I	PROD5
PROD5	$\partial(\delta p/\delta y)/\partial y$	I	A 2x64 matrix that contains the total first partials of the matrix DEPDEY wrt the OL state.	/MATS	/(310)	ALGCON	I	PROD1
						ALGCON	I	PROD5
						APPLY	I	PROD1
						APPLY	I	PROD5
PROD9		I	A 2x64 matrix that contains $\partial(\delta p/\delta y)/\partial\lambda_v$, $\partial(\delta p/\delta y)/\partial\lambda_\gamma$ and $\partial(\delta p/\delta y)/\partial\lambda_\psi$.	/MATS	/(502)	ALGCON	I	PROD9
						APPLY	I	PROB9
SFC		I		/DYNA	/(225)	APPLY	I	SFC
SFCH		I	Partial of SFC wrt h	/DYNA	/(227)	APPLY	I	SFCH
SFCHH		I	Second partial of SFC wrt h	/DYNA	/(229)	APPLY	I	SFCHH
SFCV		I	Partial of SFC wrt V	/DYNA	/(226)	APPLY	I	SFCV
SFCVH		I	Second partial of SFC wrt V and h	/DYNA	/(230)	APPLY	I	SFCVH
SFCVV		I	Second partial of SFC wrt V	/DYNA	/(228)	APPLY	I	SFCVV

969

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol		/DYNA	/(152)	AL1	I	SIDAE
							AL4	I	SIDAE
							AL6	I	SIDAE
							AL7	I	SIDAE
							AL8	I	SIDAE
							AL9	I	SIDAE
							APPLY	I	SIDAE
							CONTRL	I	SIDAE
							TH3	I	SIDAE
							UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol		/DYNA	/(9)	AL1	I	SINA
							AL4	I	SINA
							AL6	I	SINA
							AL7	I	SINA
							AL8	I	SINA
							AL9	I	SINA
							APPLY	I	SINA
							CONTRL	I	SINA
							OUTPUT	I	SINA
							TH3	I	SINA
							UT	M	SINA
SINPHI	$\sin\phi$	I	See symbol		/DYNA	/(92)	AL1	I	SINPHI
							AL4	I	SINPHI
							APPLY	I	SINPHI
							CONTRL	M	SINPHI
							OUTPUT	I	SINPHI
T	T	I	Thrust	(LBS)	/DYNA	/(42)	ALGCON	M	T
							AL1	I	T
							AL4	I	T
							AL6	I	T
							AL7	I	T
							AL8	I	T
							AL9	I	T
							APPLY	I	T
							ARCIN	O	T
							CONTRL	M	T
							DL2	I	T
							IMPULS	I	T
							OUTPUT	I	T
							TH1	I	T
							TH2	I	T
							TH3	I	T
							TH4	I	T

SUBRØUTINE
ARCEN

Purpose

ARCEN handles the interfacing of dynamic quantities between the present subarc and the subsequent subarc.

```
      SUBROUTINE ARCEN                                                  ARCEN
C                                                                       ARCEN
C                THIS ROUTINE HANDLES INTERFACING OF DYNAMIC QUANTI-    ARCEN
C                TIES BETWEEN PRESENT AND SUBSEQUENT ARCS.              ARCEN
C                                                                       ARCEN
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
C                COMPUTE TRAJECTORY TIME AT WHICH NEXT ARC BEGINS       ARCEN
      TSTART = TSTART + TAU                                             ARCEN
      RETURN                                                            ARCEN
      END                                                               ARCEN
```

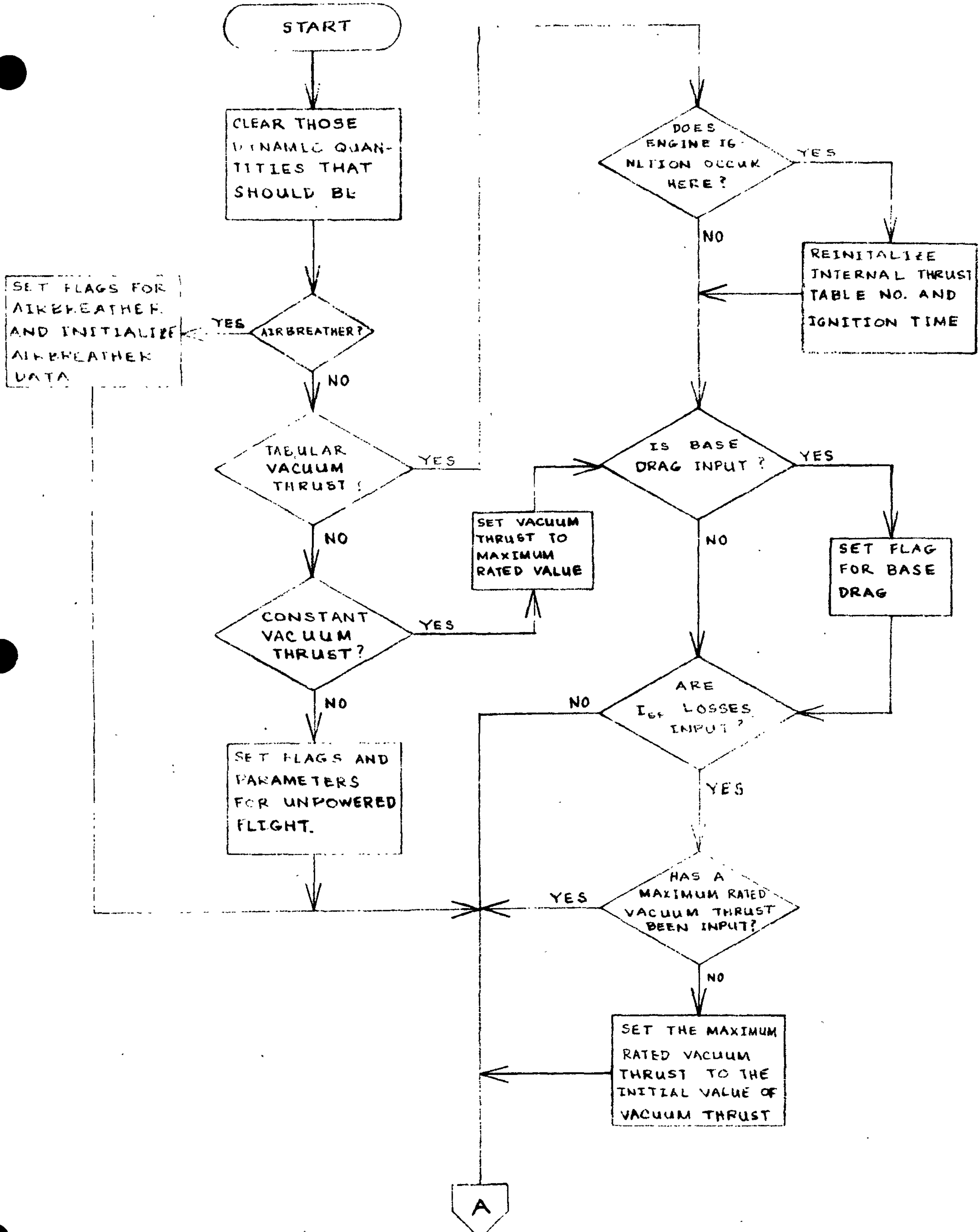
FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TAU	τ	I	Subarc duration (SEC)	/D	/(98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU
TSTART		M	Trajectory time at which present subarc commenced.	/DYNA	/(141)	ARCEN	M	TSTART
						ARCIN	M	TSTART
						STATEF	I	TSTART
						TRAJIN	O	TSTART

SUBRØUTINE ARCIN

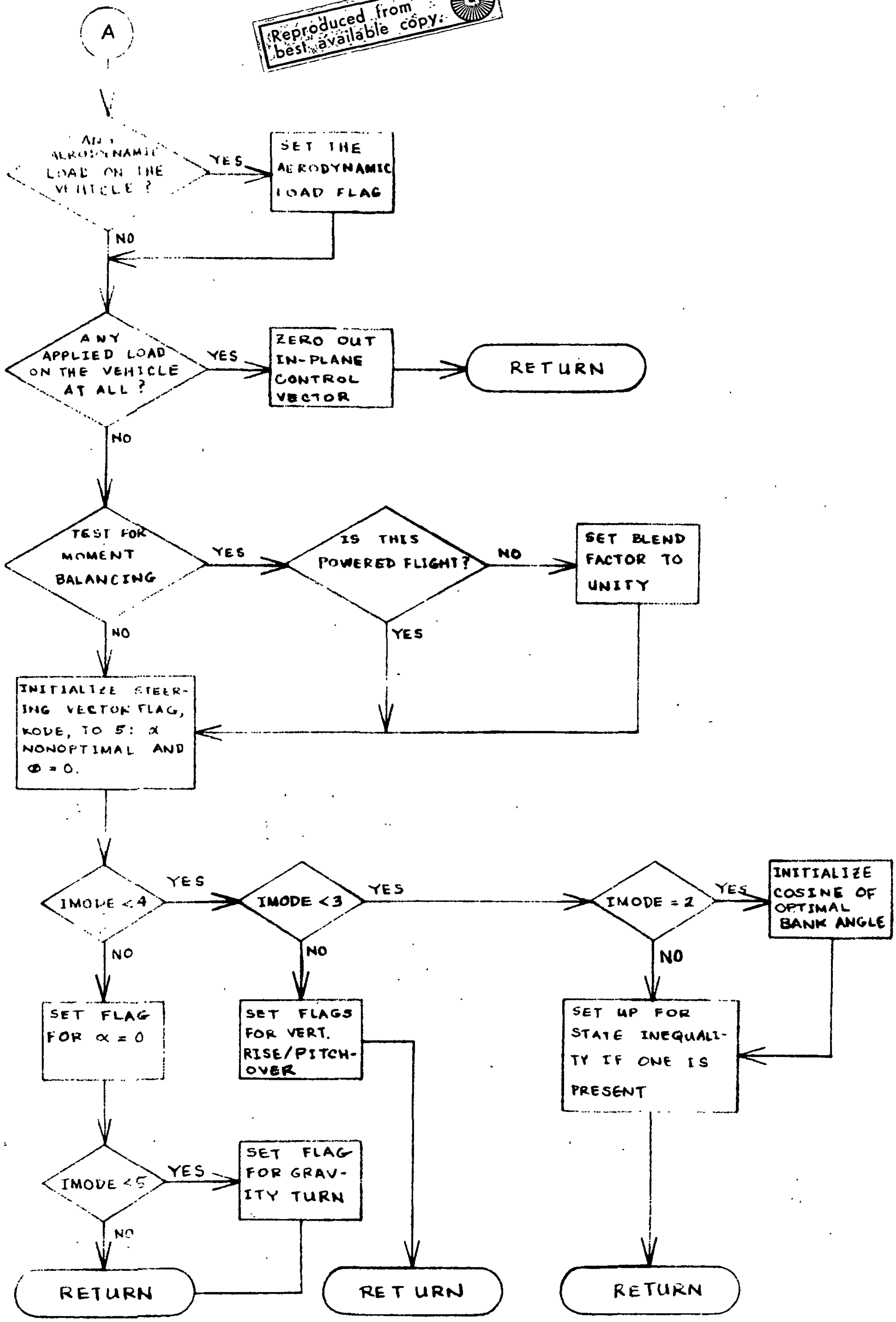
Purpose

ARCIN handles the initialization of arc-dependent flags and parameters.

ARCIN


START
CLEAR THOSE DYNAMIC QUANTITIES THAT SHOULD BE
AIRBREATHER?
YES
SET FLAGS FOR AIRBREATHER AND INITIALIZE AIRBREATHER DATA
NO
TABULAR VACUUM THRUST?
YES
NO
CONSTANT VACUUM THRUST?
YES
SET VACUUM THRUST TO MAXIMUM RATED VALUE
NO
SET FLAGS AND PARAMETERS FOR UNPOWERED FLIGHT.
DOES ENGINE IGNITION OCCUR HERE?
YES
REINITIALIZE INTERNAL THRUST TABLE NO. AND IGNITION TIME
NO
IS BASE DRAG INPUT?
YES
SET FLAG FOR BASE DRAG
NO
ARE I_{SP} LOSSES INPUT?
NO
YES
HAS A MAXIMUM RATED VACUUM THRUST BEEN INPUT?
YES
NO
SET THE MAXIMUM RATED VACUUM THRUST TO THE INITIAL VALUE OF VACUUM THRUST
A

977


A
ANY AERODYNAMIC LOAD ON THE VEHICLE ?
YES
SET THE AERODYNAMIC LOAD FLAG
NO
ANY APPLIED LOAD ON THE VEHICLE AT ALL ?
YES
ZERO OUT IN-PLANE CONTROL VECTOR
RETURN
NO
TEST FOR MOMENT BALANCING
YES
IS THIS POWERED FLIGHT?
NO
SET BLEND FACTOR TO UNITY
YES
NO
INITIALIZE STEERING VECTOR FLAG, KODE, TO 5: α NONOPTIMAL AND Φ = 0.
IMODE < 4
YES
IMODE < 3
YES
IMODE = 2
YES
INITIALIZE COSINE OF OPTIMAL BANK ANGLE
NO
SET FLAG FOR α = 0
NO
SET FLAGS FOR VERT. RISE/PITCH-OVER
NO
SET UP FOR STATE INEQUALITY IF ONE IS PRESENT
IMODE < 5
YES
SET FLAG FOR GRAVITY TURN
NO
RETURN
RETURN
RETURN

ARCIN

```
      SUBROUTINE ARCIN                                              ARCIN
C                     THIS ROUTINE HANDLES THE INITIALIZATION OF ARC-DE- ARCIN
C                     PENDENT FLAGS AND PARAMETERS.                 ARCIN
      DIMENSION XX(1)                                               ARCIN
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM D
     *, LHT                                                         D
      COMMON /D/                                                    D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20) D
      DIMENSION NOM(20)                                             D
      EQUIVALENCE (NOM, V)                                          D
      COMMON /CNTRL/                                                CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   , CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES , CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT, CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES      CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                CNTRL
      LOGICAL SWITCH, ILOAD                                         DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,  DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV, DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                           DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                             DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                    DYNA
      COMMON /DYNA/                                                 DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  , JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   , DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     , DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM , DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   , DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR , DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA , DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA , DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM , DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   , DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR, DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  , DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR, DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  , DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR, DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA, DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   , DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX , DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED  DYNA
      COMMON /DYNA/                                                 DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF , DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  , DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198, DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA, DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216, DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   , JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                             AUG09
      COMMON/ARCDAT/                                                ARCDAT
     *SREF        ,EJ          ,XISP        ,TMULT       ,DTNC        ,DTPI        , ARCDAT
     *IATM        ,IMODE       ,JAER        ,JPRO        ,QMAX        ,GMAX        , ARCDAT
     *XLMAX       ,HDMAX       ,GMDOT       ,ALFMAX      ,PHMAX       ,MAEA        , ARCDAT
     *MAEB        ,MAEC        ,MAED        ,MAEE        ,MAEF        ,MAEG        , ARCDAT
     *MT          ,MISP        ,MXCG        ,MZCG        ,MWDA        ,MWDB        , ARCDAT
     *MDB         ,XCGR        ,ZCGR        ,XE          ,ZE          ,XT          , ARCDAT
     *DREF        ,MCND        ,RHOB        ,QMULT       ,REMAX       ,FRATE         ARCDAT
      DIMENSION  ARCDA(40)                                          ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                       ARCDAT
      DIMENSION PROD1(2, 64)                                        MATS
      COMMON /MATS/                                                 MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  , MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   , MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD , MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA , MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  , MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  , MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  , MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT , MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT , MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT , MATS
```

379

```
      *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,     MATS
      *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,     MATS
      *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,     MATS
      *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT       MATS
       COMMON /MATS/                                                       MATS
      *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,     MATS
      *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,     MATS
      *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,     MATS
      *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,     MATS
      *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,     MATS
      *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,     MATS
      *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,     MATS
      *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,     MATS
      *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
      *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
      *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
      *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
      *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
      *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
      *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                MATS
       COMMON /MATS/                                                       MATS
      *DPDV(3, 8), DEPDEV(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)    MATS
       COMMON /MATS/                                                       MATS
      *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,     MATS
      *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,     MATS
      *POR   ,POO   ,PLG   ,PLP                                            MATS
       EQUIVALENCE(PROD1,PROD5)                                            MATS
       COMMON/GLOBAL/                                                      GLOBAL
      *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,     GLOBAL
      *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20),   GLOBAL
      *ITAB(20), SIG,MAXTAB,GM,PSIAF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,         GLOBAL
      *INEQFL(20), IIPSO, KSOL, INARK,KGLOBL(7)                            GLOBAL
       DATA XRCIN /6HARCIN /, RAD/57.2957795130823/                        ARCIN
C                     SAVE THOSE QUANTITIES THAT ARE NOT INITIALIZED       ARCIN
       MTTS = MTT                                                          ARCIN
       SAV2 = ALPHA                                                        ARCIN
       SAV3 = TSTART                                                       ARCIN
       SAV4 = DELTAE                                                       ARCIN
       SAV5 = TSTAGE                                                       ARCIN
C                     CLEAR THOSE QUANTITIES THAT SHOULD BE.               ARCIN
       PG = 0.                                                             ARCIN
       PLG = 0.                                                            ARCIN
       PLP = 0.                                                            ARCIN
       DO 101 I = 13, 218                                                  ARCIN
   101 XX(I) = 0.                                                          ARCIN
C                     RESTORE QUANTITIES SAVED ABOVE                       ARCIN
       IDAM =-1                                                            JUL21
       MTT = MTTS                                                          ARCIN
       ALPHA = SAV2                                                        ARCIN
       TSTART = SAV3                                                       ARCIN
       DELTAE = SAV4                                                       ARCIN
       TSTAGE = SAV5                                                       ARCIN
C                     STORE THE ARC NO. IN FLOATING PT.                    ARCIN
       XARC = IARC                                                         ARCIN
C                     CHECK FOR AIRBREATHER                                JUL21
       IF(JPRO .NE. 2) GO TO 1011                                          JUL21   1011
C                     AIRBREATHER INITIALIZATION                           JUL21
       CALL INBVPD(MWDB)                                                   JUL21
       J1 = 4                                                              JUL21
       IPOW = 1                                                            JUL21
       IF(MDB .GT. 0) IPOW = 2                                             JUL21
       GO TO 109                                                           JUL21   109
C                     WAS A NEW THRUST TABLE NO. INPUT FOR THIS ARC.       ARCIN
  1011 IF(MT .LT. 0) GO TO 102                                             JUL21   102
C                     YES.  REINITIALIZE INTERNAL THRUST TABLE NO. AND     ARCIN
C                     IGNITION TIME.                                       ARCIN
       MTT = MT                                                            ARCIN
       TSTAGE = TSTART                                                     ARCIN
C                     IS THERE A POSITIVE THRUST TABLE NO.                 ARCIN
   102 IF(MTT .LE. 0) GO TO 107                                            ARCIN   107
C                     YES.  CHECK THE INITIAL VALUE IN THE TABLE.          ARCIN
       CALL SPLINE(MTT, 0., FVAC, FVACT, FVACTT)                           ARCIN
       IF(FVAC) 103, 104, 105                                              ARCIN   104 103 105
C                     LESS THAN ZERO.  PRINT WARNING                       ARCIN
```

```
  103 CALL ERROR(XRCIN, 1, DUMMY)                                  ARCIN
      GO TO 107                                                    ARCIN
C                    EXACTLY ZERO.  SEE IF CONSTANT.               ARCIN
  104 IF(ABS(FVACT) + ABS(FVACTT) .EQ. 0.) GO TO 103               ARCIN
C                    SET FLAGS AND PARAMETERS FOR POWERED FLIGHT   ARCIN
  105 IPOW = 1                                                     ARCIN
      J1 = 2                                                       ARCIN
      AE = TMULT*EJ                                                ARCIN
C                    TEST FOR BASE DRAG TABLE.                     ARCIN
      IF(MDB .GT. 0) IPOW = 2                                      ARCIN
C                    TEST FOR ISP LOSSES.                          ARCIN
      IF(MISP .LE. 0 .OR. FRATE .GT. 0.) GO TO 106                 ARCIN
C                    ISP LOSSES INPUT BUT NOT FRATE.  PRINT WARNING AND ARCIN
C                    USE INITIAL VALUE OF FVAC.                    ARCIN
      CALL ERROR(XRCIN, 2, DUMMY)                                  ARCIN
      FRATE = FVAC                                                 ARCIN
C                    SCALE FRATE AND ISP FOR PERCENTAGES AND INITIALIZE ARCIN
C                    NET THRUST                                    ARCIN
  106 FRATED = FRATE/100.*TMULT                                    ARCIN
      IRATED = XISP/100.                                           ARCIN
      T = FVAC                                                     ARCIN
      GO TO 109                                                    ARCIN
C                    SEE IF FRATE WAS INPUT                        ARCIN
  107 IF(FRATE .LE. 0.) GO TO 108                                  ARCIN
C                    YES.  USE FRATE FOR VACUUM THRUST             ARCIN
      FVAC = TMULT*FRATE                                           ARCIN
      GO TO 105                                                    ARCIN
C                    SET FLAGS AND PARAMETERS FOR UNPOWERED FLIGHT. ARCIN
  108 J1 = 1                                                       ARCIN
      ISP = 1.                                                     ARCIN
C                    CHECK FOR AERODYNAMIC LOAD ON THE VEHICLE.    ARCIN
  109 IF(SREF .GT. 0. .AND. IATM .LT. 2) ILOAD = .TRUE.            ARCIN
C                    SEE IF THERE ARE ANY APPLIED LOADS ON THE VEHICLE ARCIN
C                    AT ALL                                        ARCIN
      IF(IPOW .NE. 0 .OR. ILOAD) GO TO 110                         ARCIN
C                    NO APPLIED LOADS.  ZERO OUT IN PLANE CONTROL AND ARCIN
C                    RETURN                                        ARCIN
      T = 0.                                                       ARCIN
      DELTAE = 0.                                                  ARCIN
      ALPHA = 0.                                                   ARCIN
      RETURN                                                       ARCIN
C                    INITIALIZE BIVARIATE AERO DATA IF CALLED FOR  ARCIN
  110 IF(ILOAD .AND. JAER .EQ. 2) CALL INBVAD                      ARCIN
C                    SET ENGINE DEFLECTION FLAG FOR ZERO ENGINE DEFLEC- ARCIN
C                    TION                                          ARCIN
      J2 = 1                                                       ARCIN
C                    TEST FOR MOMENT BALANCE                       ARCIN
      IF(JAER .NE. 3) GO TO 1103                                   ARCIN
C                    YES.  MOMENT BALANCING                        ARCIN
      IF(IPOW) 1101, 1101, 1102                                    ARCIN
C                    UNPOWERED.  SET BLEND FACTOR TO 1.            ARCIN
 1101 XJ = 1.                                                      ARCIN
      GO TO 1103                                                   ARCIN
C                    POWERED.  SET FLAG FOR NONZERO DEFLECTION     ARCIN
 1102 J2 = 2                                                       ARCIN
C                    SET STEERING FLAGS AND PARAMETERS             ARCIN
 1103 KODE = 5                                                     ARCIN
C                    CONVERT ALPHA LIMIT TO RADIANS                ARCIN
      ALMAX = ALFMAX/RAD                                           ARCIN
      IF(IMODE .LT. 4) GO TO 111                                   ARCIN
C                    GRAVITY TURN                                  ARCIN
      COSPHI = 1.                                                  ARCIN
      J3 = 2                                                       ARCIN
      IF (IMODE .LT. 5) J3 = 6                                     ARCIN
      RETURN                                                       ARCIN
  111 IF(IMODE .LT. 3) GO TO 112                                   ARCIN
C                    VERTICAL RISE OR PITCHOVER                    ARCIN
      J3 = 4                                                       ARCIN
      KODE = 4                                                     ARCIN
```

```
      GAMMAD = GMDOT/RAD                                                ARCIN
      RETURN                                                            ARCIN
  112 COSPHI = 1.                                                       ARCIN
      IF(IMODE .EQ. 2) GO TO 113                                        ARCIN
C                    OPTIMAL BANK ANGLE.  INITIALIZE COSINE OF BANK ANGLE ARCIN
      CGLG = LGAM*COS(GAM)                                              ARCIN
      COSPHI = CGLG/SQRT(LPSI**2 + CGLG**2)                             ARCIN
  113 IF(IARC .EQ. 1) RETURN                                            ARCIN
C                    SET UP FOR STATE INEQUALITY CONSTRAINT.            ARCIN
      INQF = INEQFL(IARC - 1)                                           ARCIN
      IF(INQF .EQ. 0 .OR. KONVER) RETURN                                ARCIN
      IF(INQF .EQ. 9) IDAM = 1                                          ARCIN
      RETURN                                                            ARCIN
      END                                                               ARCIN
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AE	A_{exit}	O	Total nozzle exit area	(FT^2)	/DYNA /(	89)	APPLY	I	AE
							ARCIN	O	AE
							IMPULS	I	AE
							NLDRV	I	AE
							TH2	I	AE
ALFMAX	α_{MAX}	I	Maximum angle of attack	(DEG)	/ARCDAT/(	16)	ARCIN	I	ALFMAX
							NPLANE	I	ALFMAX
ALMAX	α_{max}	O	Magnitude of angle of attack consterint	(RADS)	/DYNA /(	162)	ARCIN	O	ALMAX
							NPLANE	I	ALMAX
ALPHA	α	M	Angle of attack	(RAD)	/DYNA /(	79)	AEROCO	I	ALPHA
							ALGCON	M	ALPHA
							AL2	I	ALPHA
							ARCIN	M	ALPHA
							CONTRL	M	ALPHA
							ENVPRO	I	ALPHA
							MOMECO	I	ALPHA
							NPLANE	I	ALPHA
							OUTPUT	I	ALPHA
							TRAJIN	O	ALPHA
							UT	I	ALPHA
							WRAPUP	I	ALPHA
COSPHI	$\cos\phi$	O	See symbol		/DYNA /(	93)	AL1	I	COSPHI
							AL4	I	COSPHI
							APPLY	I	COSPHI
							ARCIN	O	COSPHI
							CONTRL	M	COSPHI
							OUTPUT	I	COSPHI
DELTAE	δ_E	M	Engine deflection	(RADS)	/DYNA /(	155)	ALGCON	M	DELTAE
							ARCIN	M	DELTAE
							CONTRL	M	DELTAE
							DL1	I	DELTAE
							OUTPUT	I	DELTAE
							TRAJIN	O	DELTAE
							UT	I	DELTAE
EJ	A_{exit}	I	Nozzle exit area	(FT^2)	/ARCDAT/(	2)	ARCIN	I	EJ
FRATE		M	Input rated vacuum thrust per engine	(LBS)	/ARCDAT/(	42)	ARCIN	M	FRATE
FRATED		O	1% of the maximum rocket vacuum thrust	(LBS)	/DYNA /(	171)	ARCIN	O	FRATED
							IMPULS	I	FRATED
FVAC		M	Total vacuum thrust (rocket)	(LBS)	/DYNA /(	33)	APPLY	I	FVAC
							ARCIN	M	FVAC
							IMPULS	M	FVAC
							NLDRV	I	FVAC
							STATEF	M	FVAC
							TH2	I	FVAC
FVACT		I	Not used.		/DYNA /(	37)	ARCIN	I	FVACT
							STATEF	M	FVACT
							TH2	I	FVACT
FVACTT		I	Not used.		/DYNA /(	41)	ARCIN	I	FVACTT
							STATEF	M	FVACTT
							TH2	I	FVACTT
GAM	γ	I	Relative flight path angle.	(RAD)	/D /(	92)	ARCIN	I	GAM
							ENVPRO	I	GAM
							OUTPUT	I	GAM
							STATEF	I	GAM
							WRAPUP	I	GAM
GAMMAD		O	Pitch rate	(RAD/SEC)	/DYNA /(	88)	AL4	I	GAMMAD
							ARCIN	O	GAMMAD
							CONTRL	I	GAMMAD
							NLDRV	I	GAMMAD
GMDOT	$\dot{\gamma}_*$	I	Pitch rate	(DEG/SEC)	/ARCDAT/(	15)	ARCIN	I	GMDOT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	[illegible]
IARC	I	I	Subarc number.		/CNTRL /(	24)	ARCIN	I	IARC
							BCOND	M	IARC
							BNDRY	M	IARC
							BRANPT	I	IARC
							CHECK	M	IARC
							COSTAB	I	IARC
							COSTAT	I	IARC
							ENDPT	I	IARC
							FORCES	I	IARC
							INARC	M	IARC
							INTRPT	I	IARC
							MAGIC	M	IARC
							MARCH	I	IARC
							OLTOSZ	I	IARC
							SALVE	M	IARC
							WRAPUP	M	IARC
IATM		I	Atmosphere option flag		/ARCDAT/(	7)	ARCIN	I	IATM
							NLORY	I	IATM
							OUTPUT	I	IATM
							STATEF	I	IATM
IDAM		O	Optional atmospheric calculations flag. IDAM = -1: Compute $\partial^3\rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3\rho_a/\partial R^3$, μ_a, $\partial\mu_a/\partial R$, etc.		/DYNA /(	218)	ARCIN	O	IDAM
							ERROR	I	IDAM
							NPLANE	O	IDAM
							STATEF	I	IDAM
							WRAPUP	O	IDAM
ILOAD		M	Logical flag that is true if there is any aerodynamic load on the vehicle.		/DYNA /(	181)	ARCIN	M	ILOAD
							CONTRL	I	ILOAD
							NPLANE	I	ILOAD
							UT	I	ILOAD
IMODE		I	Control mode option flag		/ARCDAT/(	8)	ARCIN	I	IMODE
							CONTRL	I	IMODE
							NPLANE	I	IMODE
INEQFL		I	A 20 word array that contains the code number of the state variables inequality constraint that applies on each subarc. A zero entry indicates that no SVIC applies.		/GLOBAL/(	72)	ARCIN	I	INEQFL
INQF		M	State variable inequality constraint flag. INQF = 0: No SVIC in effect; INQF = 7: Dynamic pressure IC in effect; INQF = 8: Heating rate SVIC in effect; INQF = 9: Reynolds number SVIC in effect.		/DYNA /(	185)	ARCIN	M	INQF
							NPLANE	M	INQF
IPOW		M	Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag		/DYNA /(	139)	ARCIN	M	IPOW
							FORCES	I	IPOW
							NPLANE	I	IPOW
							STATEF	I	IPOW
							THROTL	I	IPOW
IRATED		O	1% of the maximum rated I_{sp}	(SECS)	/DYNA /(	170)	ARCIN	O	IRATED
							IMPULS	I	IRATED
ISP	I_{sp}	O	Vacuum specific impulse	(SECS)	/DYNA /(	45)	APPLY	I	ISP
							ARCIN	O	ISP
							IMPULS	O	ISP
JAER		I	Aerodynamic model option flag		/ARCDAT/(	9)	AEROCD	I	JAER
							ARCIN	I	JAER
							OUTPUT	I	JAER
							STATEF	I	JAER
							UT	I	JAER
JPRO		I	Propulsion model option flag		/ARCDAT/(	10)	ARCIN	I	JPRO
							IMPULS	I	JPRO
J1		O	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.		/DYNA /(	173)	APPLY	I	J1
							ARCIN	O	J1
							CONTRL	M	J1
							FORCES	I	J1
							NPLANE	I	J1
							STATEF	I	J1
							THROTL	M	J1

984

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
J2		O	Engine deflection option flag. J2 = 1: Constant engine deflection; J2 = 2: Moment balancing.	/DYNA /(	174)	ARCIN CONTRL NPLANE	O I I	J2 J2 J2
J3		O	Angle of attack option flag. J3 = 1: Optimal angle of attack; J3 = 2: Constant angle of attack; J3 = 3: Untrimmed lift limit; J3 = 4: Vertical rise or pitchover; J3 = 5: Unpowered total acceleration limit; J3 = 6: Gravity turn; J3 = 7: Dynamic pressure constraint; J3 = 8: Heating rate constraint; J3 = 9: Reynolds number constraint.	/DYNA /(	175)	ARCIN CONTRL NPLANE OUTPUT	O M M I	J3 J3 J3 J3
KODE		O	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.	/DYNA /(	25)	APPLY ARCIN CONTRL FORCES NLDRV STATEF	I O M I I I	KODE KODE KODE KODE KODE KODE
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON APPLY ARCIN COMOMO GROPE NLDRV OUTPUT RKUTT1	I I I O O I I I	KONVER KONVER KONVER KONVER KONVER KONVER KONVER KONVER
LGAM	λ_γ	I	Relative flight path angle costate	/D /(	101)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LGAM LGAM LGAM LGAM LGAM LGAM
LPSI	λ_ψ	I	Relative azimuth angle costate	/D /(	102)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LPSI LPSI LPSI LPSI LPSI LPSI
MDB		I	Curve number - base drag table	/ARCDAT/(	31)	ARCIN STATEF	I I	MDB MDB
MISP		I	Curve number KISP loss table	/ARCDAT/(	26)	ARCIN IMPULS	I I	MISP MISP
MT		I	Curve number -thrust table	/ARCDAT/(	25)	ARCIN	I	MT
MTT		M	Table number for tabulated rocket vacuum thrust	/DYNA /(	172)	ARCIN STATEF	M I	MTT MTT
PG	ϕ_γ	O	See symbol	/MATS /(	551)	AL4 APPLY ARCIN CONTRL	M I O O	PG PG PG PG
PLG	ϕ_{λ_γ}	O	See symbol	/MATS /(	570)	APPLY ARCIN CONTRL	I O O	PLG PLG PLG
PLP	ϕ_{λ_ψ}	O	See symbol	/MATS /(	571)	APPLY ARCIN CONTRL	I O O	PLP PLP PLP
SREF	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	ARCIN BNDRY CHECK FETCH SALVE STATEF UT WRAPUP	I I I I I I I I	SREF ARCDA ARCDA ARCDA ARCDA SREF SREF ARCDA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
T	T	O	Thrust (LBS)	/DYNA /(	42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
TMULT	T_{mult}	I	Thrust multiplier or number of engines	/ARCDAT/(	4)	ARCIN	I	TMULT
						STATEF	I	TMULT
TSTAGE		M	Trajectory time at which present rocket engine ignited. (SECS)	/DYNA /(	167)	ARCIN	M	TSTAGE
						STATEF	I	TSTAGE
						TRAJIN	O	TSTAGE
TSTART		M	Trajectory time at which present subarc commenced.	/DYNA /(	141)	ARCEN	M	TSTART
						ARCIN	M	TSTART
						STATEF	I	TSTART
						TRAJIN	O	TSTART
XARC		O	Quasitime at which present subarc commenced.	/DYNA /(	140)	ARCIN	O	XARC
						STATEF	I	XARC
XISP	I_{sp}	I	Vacuum specific impulse (SEC)	/ARCDAT/(	3)	ARCIN	I	XISP
						IMPULS	I	XISP
XJ	$\dot{j}$	O	Control blend factor	/DYNA /(	159)	ARCIN	O	XJ
						DL2	I	XJ
						OUTPUT	I	XJ
						STATEF	I	XJ
						UT	I	XJ
XX		O	Fraction of subarc that has transpired	/DYNA /(	1)	ARCIN	O	XX
						ERROR	I	XX
						OUTPUT	I	XX
						STATEF	M	XX

Purpose

Given the matrix A in Equation 16.6-29 in Vol. I, BASIS determines the matrix A* in Equation 16.6-33 in Vol. I.

```
      SUBROUTINE BASIS(A, M, N)                                          BASIS
C                                                                        BASIS
C                     GIVEN A SET OF M LINEARLY INDEPENDENT ROW VECTORS IN BASIS
C                     THE EUCLIDEAN SPACE E**N, WHICH ARE STORED IN THE   BASIS
C                     FIRST M ROWS OF THE MATRIX A, THIS ROUTINE GENERATES BASIS
C                     A SET OF N LINEARLY INDEPENDENT ROW VECTORS IN E**N. BASIS
C                                                                        BASIS
      DIMENSION A(N, N), LNZE(30)                                        BASIS
C                     ROW REDUCE THE FIRST M ROWS OF A                   BASIS
      DO 105 I = 1, M                                                    BASIS
      DO 104 J = 1, N                                                    BASIS
C                     FIND LEADING NON-ZERO ENTRY IN I-TH ROW            BASIS
      IF(ABS(A(I, J)).LE. 0.) GO TO 104                                  BASIS
C                     STORE COLUMN NUMBER IN WHICH LEADING NON-ZERO OCCURS BASIS
      LNZE(I) = J                                                        BASIS
C                     NORMALIZE I-TH ROW BY LEADING NON-ZERO ENTRY       BASIS
      FNZE = A(I, J)                                                     BASIS
      DO 101 K = 1, N                                                    BASIS
  101 A(I, K) = A(I, K)/FNZE                                             BASIS
      A(I, J) = 1                                                        BASIS
C                     ANNIHILATE NON-ZERO ENTRIES ABOVE AND BELOW LEADING BASIS
C                     NON-ZERO.                                          BASIS
      DO 103 K = 1, M                                                    BASIS
      IF(K .EQ. I) GO TO 103                                             BASIS
      IF(ABS(A(K, J)) .LE. 0.) GO TO 103                                 BASIS
      AKJ = A(K, J)                                                      BASIS
      DO 102 L = 1, N                                                    BASIS
  102 A(K, L) = A(K, L) - AKJ*A(I, L)                                    BASIS
      A(K, J) = 0                                                        BASIS
  103 CONTINUE                                                           BASIS
      GO TO 105                                                          BASIS
  104 CONTINUE                                                           BASIS
  105 CONTINUE                                                           BASIS
C                     GENERATE REMAINING N - M ROW VECTORS BY STORING A  BASIS
C                     ONE IN THOSE COLUMNS THAT DO NOT CONTAIN A LEADING BASIS
C                     NON-ZERO ENTRY.                                    BASIS
      L = M + 1                                                          BASIS
      MA = M                                                             BASIS
      DO 108 I = L, N                                                    BASIS
      DO 107 J = 1, N                                                    BASIS
      DO 106 K = 1, MA                                                   BASIS
  106 IF(LNZE(K) .EQ. J) GO TO 107                                       BASIS
      A(I, J) = 1                                                        BASIS
      LNZE(I) = J                                                        BASIS
      MA= MA+ 1                                                          BASIS
      GO TO 108                                                          BASIS
  107 CONTINUE                                                           BASIS
  108 CONTINUE                                                           BASIS
      RETURN                                                             BASIS
      END                                                                BASIS
```

SUBRØUTINE
BCØND

Purpose

BCOND sets up the state initial conditions block IIC and the state target conditions blocks ITC and JTAB.

```
      SUBROUTINE BCOND                                                  BCOND
C                                                                       BCOND
C                 THIS ROUTINE SETS UP THE STATE INITIAL CONDITIONS     BCOND
C                 BLOCK IIC AND THE STATE TARGET CONDITIONS BLOCKS      BCOND
C                 ITC AND JTAB                                          BCOND
C                                                                       BCOND
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LB, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      DIMENSION CONDS(400), FIC(1), FTC(1), FTRNS(1), INTR(17),         BCOND
     *IRTS(17), ISD(17), ISTATE(12), ITRNS(24), JTC(12), KTC(12),       BCOND
     *MTC(1), VTC(1), TABLE(1)                                          BCOND
      EQUIVALENCE (FIC, CONDS), (FTC, CONDS(301)), (VTC, ITC(51)),      BCOND
     *(MTC, ITC(101)), (ITRNS, FTRNS), (NRTS, IRTS), (ITABLE, TABLE)    BCOND
      DATA NS/9/, INTR/8, 1, 2, 4, 7, 3, 5, 6,0,9,0,21,0, 24, 29, 2*0/, BCOND
     *ISD/128, 1, 2, 8, 64, 4, 16, 32,0,256,0,31,0,31, 48, 2*0/,        BCOND
     *IALL/511/, ISTATE/1, 2, 4, 8, 16, 32, 64, 128, 256, 3*0/,         BCOND
     *FLSHFT/4.096E3/,                                                  BCOND
     *IRTS/6, 3, 86037, 98325, 438336, 2, 86037, 98325, 1, 524336, 1,   BCOND
     *413697, 2, 98324, 438336, 1, 438336/,                             BCOND
     *ITRNS/2, 25, 0, 16, 1., 1, 25, 0, 2, 26, 0, 27, 0, 1,10,1. ,2,    BCOND
     *28, 0, 16, 1., 1, 16, 1./,                                        BCOND
     *XBCNDS/6H BCOND/                                                  BCOND
      IARC = 1                                                          BCOND
C                SET INTEGRATION CONTROL PARAMETERS SO THAT SUBROU-     BCOND
C                TINES ARCIN AND ARCEN WILL BE CALLED BY FORCES WHEN    BCOND
C                FETCH CALL FORCES.                                     BCOND
C                                                                       BCOND
      KPT = 1                                                           BCOND
      NPTS = 1                                                          BCOND
C                READ INITIAL AND TARGET CONDITIONS INTO ARRAY CONDS    BCOND
      CALL READMS(9, CONDS, 400, 21)                                    BCOND
C                CLEAR STATE INITIAL AND TARGET CONDITION BLOCKS        BCOND
      DO 101 I = 1, 9                                                   BCOND
      DO 101 J = 1, NARC                                                BCOND
      IIC(I, J) = 0                                                     BCOND
      ITC(I, J) = 0                                                     BCOND
      VALIC(I, J) = 0                                                   BCOND
  101 VALTC(I, J) = 0                                                   BCOND
C                GET INITIAL VALUE OF STATE AND COSTATE OFF THE INI-    BCOND
C                TIAL ARC FILE                                          BCOND
      CALL FETCH(0)                                                     BCOND
C                STORE INITIAL STATE AND COSTATE                        BCOND
      DO 102 I = 1, 18                                                  BCOND
  102 ZSAVE(I) = NOM(I)                                                 BCOND
C                INITIALIZE FIRST COLUMN OF IIC TO KNOWN CONDITION      BCOND
      DO 103 I = 1, 9                                                   BCOND
  103 IIC(I, 1) = 1                                                     BCOND
C                INTERPRET INPUT CONDITIONS                             BCOND
      INDX = -2                                                         BCOND
      JNDX = -1                                                         BCOND
      DO 109 IARC = 1, NARC                                             BCOND
C                STORE NUMBER OF STATED INITIAL CONDITIONS FOR THIS     BCOND
C                ARC                                                    BCOND
```

```
      NOSIC = ITAB(IARC)                                                BCOND
      IF(NOSIC .LE. 0) GO TO 105                                        BCOND
C                   INTERPRET THE STATED INITIAL CONDITIONS             BCOND
      DO 104 I= 1, NOSIC                                                BCOND
      INDX = INDX + 3                                                   BCOND
      ICODE = FIC(INDX) + .5                                            BCOND
C                   IF INIT. COND. CODE .GE. 10 ASSUME STATE CONTINUOS  BCOND
      IF(ICODE .GE. 10) ICODE = 0                                       BCOND
      JVN = FIC(INDX +1) + .5                                           BCOND
C                   TRANSLATE STEEP. DESC. VARIABLE NO. TO OL NO.       BCOND
      IVN = INTR(JVN)                                                   BCOND
C                   SKIP INIT. COND. ON TRAJ. TIME                      BCOND
      IF(IVN .EQ. 0) GO TO 104                                          BCOND
      IIC(IVN, IARC) = ICODE                                            BCOND
C                   CONVERT DESIRED VALUE TO INTERNAL UNITS             BCOND
      VALIC(IVN, IARC) = SOMG(JVN, FIC(INDX + 2))                       BCOND
  104 CONTINUE                                                          BCOND
C                   STORE NUMBER OF TARGET CONDS. FOR THIS ARC          BCOND
  105 NMBR = KTAB(IARC)                                                 BCOND
      L = 0                                                             BCOND
      IF(NMBR .LE. 0) GO TO 108                                         BCOND
C                   INTERPRET TARGET CONDS.                             BCOND
      DO 107 I = 1, NMBR                                                BCOND
      JNDX = JNDX + 2                                                   BCOND
      FWRD = ABS(FTC(JNDX))                                             BCOND
      IWRD = 1.E-6*FWRD + .5                                            BCOND
      GWRD = IWRD                                                       BCOND
      JWRD = FWRD - 1.E6*GWRD + .5                                      BCOND
C                   CHECK FOR PAYOFF CONDITION                          BCOND
      IF(JWRD .LT. 2) GO TO 106                                         BCOND
      IPAY = IWRD                                                       BCOND
      GO TO 107                                                         BCOND
  106 L = L + 1                                                         BCOND
      ITC(L, IARC) = IWRD                                               BCOND
C                   CONVERT DESIRED VALUE TO INTERNAL UNITS             BCOND
      VALTC(L, IARC) = SOMG(IWRD, FTC(JNDX + 1))                        BCOND
  107 CONTINUE                                                          BCOND
C                   STORE ACTUAL NUMBER OF TARGETS IN JTAB              BCOND
  108 JTAB(IARC) = L                                                    BCOND
C                   MAKE SURE PAYOFF WAS ON LAST ARC                    BCOND
  109 IF(L .LT. NMBR .AND. IARC .NE. NARC) CALL ERROR(XBCNDS, -1, 1)    BCOND
C                   SUPERIMPOSE INPUT INITIAL GROSS MASS                BCOND
      IF(IIC(7, 1) .EQ. 1) ZSAVE(7) = VALIC(7, 1)                       BCOND
C                   DETERMINE THE COSTATE INITIAL AND TARGET CONDITIONS BCOND
      CALL MAGIC                                                        BCOND
C                   STORE TRANSVERSALITY CONDITIONS FOR END OF LAST     BCOND
C                   BRANCH                                              BCOND
      MA = 9 - JTAB(NARC)                                               BCOND
      DO 110 I = 1, MA                                                  BCOND
  110 ITCT(I, NARC) = I                                                 BCOND
      LTAB(NARC) = MA                                                   BCOND
C                   TEST FOR BRANCH PROBLEM                             BCOND
      IF(NFARC .EQ. NARC) RETURN                                        BCOND
C                   STORE TRANSVERSALITY CONDITIONS FOR END OF FIRST    BCOND
C                   BRANCH                                              BCOND
      MA = 9 - JTAB(NFARC)                                              BCOND
      DO 111 I = 1, MA                                                  BCOND
  111 ITCT(I, NFARC) = I                                                BCOND
      LTAB(NFARC) = MA                                                  BCOND
      RETURN                                                            BCOND
      END                                                               BCOND
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
IARC	I	M	Subarc number.	/CNTRL /(	24)	ARCIN BCOND BNDRY BRANPT CHECK COSTAB COSTAI ENDPT FORCES INARC INTRPT MAGIC MARCH QLTOSZ SALVE WRAPUP	I M M I M I I I I M I M I I M M	IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC
IIC		M	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND BRANPT CHECK COSTAB COSTAI COSTAO INTRPT SALVE	M I I I I I I I	IIC IIC IIC IIC IIC IIC IIC IIC
INDX		M	An array of four words that indicate to Adams-Moulton integration in what order the derivatives of the particular and homogeneous solutions are stored.	/CNTRL /(	11)	BCOND MADAMS SALVE	M M O	INDX INDX INDX
ITAB		I	A 20 word array containing the number of nonzero state initial conditions specified at the beginning of each subarc.	/GLOBAL/(	45)	BCOND	I	ITAB
ITC		O	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT	O I I I I I I	ITC ITC ITC ITC ITC ITC ITC
ITCT		O	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND BRANPT CHECK COSTAB COSTAI INTRPT MAGIC	O I I O O I O	ITCT ITCT ITCT ITCT ITCT ITCT ITCT
JTAB		M	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT MAGIC	M I I I I I I I	JTAB JTAB JTAB JTAB JTAB JTAB JTAB JTAB
KPT		O	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND BNDRY FORCES MAGIC RKUTT1 SALVE WRAPUP	O O I O I M M	KPT KPT KPT KPT KPT KPT KPT
KTAB		I	A 20 word array containing the number of state target conditions specified at the end of each subarc.	/GLOBAL/(	25)	BCOND	I	KTAB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LTAB		O	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND	O	LTAB
						BRANPT	I	LTAB
						COSTAB	O	LTAB
						COSTAI	O	LTAB
						INTRPT	I	LTAB
						MAGIC	M	LTAB
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRO	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						OLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRO	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						OLTOSZ	I	NFARC
						SALVE	I	NFARC
NOM	V	I	Relative velocity. (FT/SEC)	/D /(	91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
NPTS		O	The total number of points in the subarc.	/CNTRL /(	19)	BCOND	O	NPTS
						BNDRY	O	NPTS
						FORCES	I	NPTS
						INARC	M	NPTS
						MAGIC	O	NPTS
						SALVE	M	NPTS
						WRAPUP	O	NPTS
VALIC		M	A 10x20 array containing the desired values of all the fixed (known) QL state variables. The columns correspond to the subarc starting points, the rows, to QL state variables.	/BLOCK /(	862)	BCOND	M	VALIC
						SALVE	I	VALIC
VALTC		O	A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK /(	1062)	BCOND	O	VALTC
						BRANPT	I	VALTC
						CHECK	I	VALTC
						ENDPT	I	VALTC
						INTRPT	I	VALTC
ZSAVE		O	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D /(	151)	BCOND	O	ZSAVE
						BRANPT	I	ZSAVE
						COSTAB	I	ZSAVE
						COSTAI	I	ZSAVE
						INTRPT	I	ZSAVE
						PDBCOL	I	ZSAVE
						SALVE	I	ZSAVE

SUBRØUTINE BLINE

Purpose

BLINE interpolates the bivariate tabular functions. In addition, it computes the first and second partials of these functions by evaluating the derivatives of the bicubic spline interpolating function.*

*See Section 17.7 of Vol. I.

```
      SUBROUTINE BLINE(A, M, U)                                         BLINE
C                                                                       BLINE
C                    THIS ROUTINE COMPUTES THE BIVARIATE AERODYNAMIC    BLINE
C                    LIFT AND DRAG COEFFICIENTS AND THEIR FIRST AND     BLINE
C                    SECOND PARTIALS.  A IS THE CURRENT ANGLE OF ATTACK, BLINE
C                    M IS MACH AND U = CL, CLA, CLM, CLAA, CLMM, CLAM,  BLINE
C                    CD, CDA, CDM, CDAA, CDMM, CDAM                     BLINE
C                                                                       BLINE
      REAL M, K, MMIN, MMAX, MACH                                       BLINE
      COMMON /BICUBE/ AMIN, AMAX, IF, IFMAX, MMIN, MMAX, IR, IRMAX,     BLINE
     *IUNIT, IRECT, IREC, C(32), T(160), KNOTS(1)                       BLINE
      DIMENSION U(12), ALFA(1), MACH(1)                                 BLINE
      EQUIVALENCE (ALFA, KNOTS), (MACH, KNOTS(32))                      BLINE
      EQUIVALENCE                                                       BLINE
     *(CL00, C( 1)), (CL01, C( 5)), (CL02, C( 9)), (CL03, C(13)),       BLINE
     *(CL10, C( 2)), (CL11, C( 6)), (CL12, C(10)), (CL13, C(14)),       BLINE
     *(CL20, C( 3)), (CL21, C( 7)), (CL22, C(11)), (CL23, C(15)),       BLINE
     *(CL30, C( 4)), (CL31, C( 8)), (CL32, C(12)), (CL33, C(16)),       BLINE
     *(CD00, C(17)), (CD01, C(21)), (CD02, C(25)), (CD03, C(29)),       BLINE
     *(CD10, C(18)), (CD11, C(22)), (CD12, C(26)), (CD13, C(30)),       BLINE
     *(CD20, C(19)), (CD21, C(23)), (CD22, C(27)), (CD23, C(31)),       BLINE
     *(CD30, C(20)), (CD31, C(24)), (CD32, C(28)), (CD33, C(32))        BLINE
C                    SEE IF MACH IS WITHIN RANGE                        BLINE
      IF(MMIN .LE. M .AND. M .LE. MMAX) GO TO 101                       BLINE
  100 CONTINUE                                                          BLINE
C                    SEE IF ALPHA IS WITHIN RANGE                       BLINE
  101 IF(AMIN .LE. A .AND. A .LE. AMAX) GO TO 103                       BLINE
C                    NO.  FORCE ALPHA TO APPROPRIATE LIMIT.             BLINE
      IF(A .LT. AMIN) A = AMIN                                          BLINE
      IF(AMAX .LT. A) A = AMAX                                          BLINE
C                    FIND LARGEST MACH ENTRY .LE. CURRENT MACH NO.      BLINE
  103 IF(M - MACH(IR)) 104, 109, 105                                    BLINE
  104 IR = IR - 1                                                       BLINE
      GO TO 103                                                         BLINE
  105 IF(M - MACH(IR + 1)) 109, 106, 107                                BLINE
  106 IF(IR .GE. IRMAX) GO TO 109                                       BLINE
  107 IR = IR + 1                                                       BLINE
      GO TO 103                                                         BLINE
  109 IF(A - ALFA(IF)) 110, 114, 111                                    BLINE
  110 IF = IF - 1                                                       BLINE
      GO TO 109                                                         BLINE
C                    FIND LARGEST ALPHA ENTRY .LE. CURRENT ALPHA        BLINE
  111 IF(A - ALFA(IF + 1)) 114, 112, 113                                BLINE
  112 IF(IF .GE. IFMAX) GO TO 114                                       BLINE
  113 IF = IF + 1                                                       BLINE
      GO TO 109                                                         BLINE
C                    COMPUTE DISTANCES FROM GRID POINTS                 BLINE
  114 H = A - ALFA(IF)                                                  BLINE
      K = M - MACH(IR)                                                  BLINE
C                    FIND APPROPIATE GRID RECTANGLE AND STORE ITS SPLINE BLINE
C                    COEFFICIENTS IN C ARRAY.  NOTE EACH RECORD CONTAINS BLINE
C                    FIVE RECTANGLES                                    BLINE
      JRECT = IR + IRMAX*(IF - 1)                                       BLINE
      IF(JRECT .EQ. IRECT) GO TO 117                                    BLINE
      IRECT = JRECT                                                     BLINE
      JREC = (IRECT - 1)/5 + 2                                          BLINE
      IF(JREC .EQ. IREC) GO TO 115                                      BLINE
      IREC = JREC                                                       BLINE
      CALL READMS(IUNIT, T, 160, IREC)                                  BLINE
  115 IB = 32*(IRECT - 5*IREC + 9)                                      BLINE
      DO 116 I = 1, 32                                                  BLINE
      J = I + IB                                                        BLINE
  116 C(I) = T(J)                                                       BLINE
  117 CONTINUE                                                          BLINE
      CL0 = CL00 + K*(CL01 + K*(CL02 + K*CL03))                         BLINE
      CL1 = CL10 + K*(CL11 + K*(CL12 + K*CL13))                         BLINE
      CL2 = CL20 + K*(CL21 + K*(CL22 + K*CL23))                         BLINE
      CL3 = CL30 + K*(CL31 + K*(CL32 + K*CL33))                         BLINE
```

```
C                    COMPUTE CL                                        BLINE
      U(1) = CL0 + H*(CL1 + H*(CL2 + H*CL3))                           BLINE
      CD0 = CD00 + K*(CD01 + K*(CD02 + K*CD03))                        BLINE
      CD1 = CD10 + K*(CD11 + K*(CD12 + K*CD13))                        BLINE
      CD2 = CD20 + K*(CD21 + K*(CD22 + K*CD23))                        BLINE
      CD3 = CD30 + K*(CD31 + K*(CD32 + K*CD33))                        BLINE
C                    COMPUTE CD                                        BLINE
      U(7) = CD0 + H*(CD1 + H*(CD2 + H*CD3))                           BLINE
      CL0PP = K*CL03                                                   BLINE
      CL0PP = CL0PP + CL0PP + CL0PP + CL02                             BLINE
      CL0P = CL01 + K*(CL02 + CL0PP)                                   BLINE
      CL1PP = K*CL13                                                   BLINE
      CL1PP = CL1PP + CL1PP + CL1PP + CL12                             BLINE
      CL1P = CL11 + K*(CL12 + CL1PP)                                   BLINE
      CL2PP = K*CL23                                                   BLINE
      CL2PP = CL2PP + CL2PP + CL2PP + CL22                             BLINE
      CL2P = CL21 + K*(CL22 + CL2PP)                                   BLINE
      CL3PP = K*CL33                                                   BLINE
      CL3PP = CL3PP + CL3PP + CL3PP + CL32                             BLINE
      CL3P = CL31 + K*(CL32 + CL3PP)                                   BLINE
      UXX = H*CL3                                                      BLINE
      UXX = UXX + UXX + UXX + CL2                                      BLINE
C                    COMPUTE CLA AND CLM                               BLINE
      U(2) = CL1 + H*(CL2 + UXX)                                       BLINE
      U(3) = CL0P + H*(CL1P + H*(CL2P + H*CL3P))                       BLINE
      CD0PP = K*CD03                                                   BLINE
      CD0PP = CD0PP + CD0PP + CD0PP + CD02                             BLINE
      CD0P = CD01 + K*(CD02 + CD0PP)                                   BLINE
      CD1PP = K*CD13                                                   BLINE
      CD1PP = CD1PP + CD1PP + CD1PP + CD12                             BLINE
      CD1P = CD11 + K*(CD12 + CD1PP)                                   BLINE
      CD2PP = K*CD23                                                   BLINE
      CD2PP = CD2PP + CD2PP + CD2PP + CD22                             BLINE
      CD2P = CD21 + K*(CD22 + CD2PP)                                   BLINE
      CD3PP = K*CD33                                                   BLINE
      CD3PP = CD3PP + CD3PP + CD3PP + CD32                             BLINE
      CD3P = CD31 + K*(CD32 + CD3PP)                                   BLINE
      VXX = H*CD3                                                      BLINE
      VXX = VXX + VXX + VXX + CD2                                      BLINE
C                    COMPUTE CDA AND CDM                               BLINE
      U(8) = CD1 + H*(CD2 + VXX)                                       BLINE
      U(9) = CD0P + H*(CD1P + H*(CD2P + H*CD3P))                       BLINE
C                    COMPUTE CLAA, CLMM AND CLAM                       BLINE
      U(4) = UXX + UXX                                                 BLINE
      UYY = CL0PP + H*(CL1PP + H*(CL2PP + H*CL3PP))                    BLINE
      U(5) = UYY + UYY                                                 BLINE
      U(6) = CL1P + H*(CL2P + CL2P + H*(CL3P + CL3P + CL3P))           BLINE
C                    COMPUTE CDAA, CDMM AND CDAM                       BLINE
      U(10)= VXX + VXX                                                 BLINE
      VYY = CD0PP + H*(CD1PP + H*(CD2PP + H*CD3PP))                    BLINE
      U(11)= VYY + VYY                                                 BLINE
      U(12)= CD1P + H*(CD2P + CD2P + H*(CD3P + CD3P + CD3P))           BLINE
      RETURN                                                           BLINE
      END                                                              BLINE
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ALFA		I	A 31 word array containing the mesh $x_0, x_1, \ldots, x_n$	/BICUBE/(	204)	BLINE	I	ALFA
AMAX	x_N	I	The largest value of the first independent variable of a bivariate table.	/BICUBE/(	2)	BLINE	I	AMAX
AMIN	x_0	I	The smallest value of the first independent variable of a bivariate table.	/BICUBE/(	1)	BLINE	I	AMIN
C		O	A 32 word array containing the spline coefficients for the two bivariate functions at rectangle IRECT.	/BICUBE/(	12)	BLINE BLINE	O I	C CLOO
CLOO		I	A 32 word array containing the spline coefficients for the two bivariate functions at rectangle IRECT.	/BICUBE/(	12)	BLINE BLINE	O I	C CLOO
IF		M	Last file in the grid in which interpolation occurred.	/BICUBE/(	3)	BLINE	M	IF
IFMAX	N	I	Total number of files in grid.	/BICUBE/(	4)	BLINE	I	IFMAX
IR		M	Last rank in the grid in which interpolation occurred.	/BICUBE/(	7)	BLINE	M	IR
IREC		M	Logical record on IUNIT that contains spline coefficients for rectangle IRECT.	/BICUBE/(	11)	BLINE	M	IREC
IRECT		M	Grid rectangle associated with IR and IF.	/BICUBE/(	10)	BLINE	M	IRECT
IRMAX		I	Total number of ranks in grid.	/BICUBE/(	8)	BLINE	I	IRMAX
IUNIT		I	Logical unit number on which bicubic spline coefficients are stored for this table.	/BICUBE/(	9)	BLINE	I	IUNIT
MACH		I	A 31 word array containing the mesh $y_0, y_1, \ldots, y_n$	/BICUBE/(	235)	BLINE	I	MACH
MMAX	y_M	I	The largest value of the second independent variable of a bivariate table.	/BICUBE/(	6)	BLINE	I	MMAX
MMIN	y_0	I	The smallest value of the second independent variable of a bivariate table.	/BICUBE/(	5)	BLINE	I	MMIN
T		I	A 160 word array containing logical record IREC.	/BICUBE/(	44)	BLINE	I	T

SUBRØUTINE BNDRY

Purpose

BNDRY controls the computation of the state and costate target misses and the partials of these misses with respect to the c's.*

*See Sections 16.6 and 17.4 of Vol. I.

```
      SUBROUTINE BNDRY (PZ, DPZIDC, K)                                  BNDRY
C                                                                       BNDRY
C                     THIS ROUTINE CONTROLS THE COMPUTATION OF THE STATE BNDRY
C                     AND COSTATE TARGET MISSES AND THE PARTIALS OF THOSE BNDRY
C                     MISSES WITH RESPECT TO THE MULTIPLIERS OF THE HOMO- BNDRY
C                     GENEOUS SOLUTIONS, OR C*S FOR SHORT.  THE PARTIALS BNDRY
C                     ARE STORED IN DPZIDC.  THE MISSES, IN PZ.          BNDRY
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF    ,EJ      ,XISP    ,TMULT   ,DTNC    ,DTPI    ,            ARCDAT
     *IATM    ,IMODE   ,JAER    ,JPRO    ,QMAX    ,GMAX    ,            ARCDAT
     *XLMAX   ,HDMAX   ,GMDOT   ,ALFMAX  ,PHMAX   ,MAEA    ,            ARCDAT
     *MAEB    ,MAEC    ,MAED    ,MAEE    ,MAEF    ,MAEG    ,            ARCDAT
     *MT      ,MISP    ,MXCG    ,MZCG    ,MWDA    ,MWDB    ,            ARCDAT
     *MDB     ,XCGR    ,ZCGR    ,XE      ,ZE      ,XT      ,            ARCDAT
     *DREF    ,MCND    ,RHOB    ,QMULT   ,REMAX   ,FRATE                ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      COMMON /EVAL/ SGN, SPART(18), MAP(10), PZI(40), NOCK, S(18, 41),  EVAL
     *TEMP(40), DZ(18), DC, L, SI(18, 41)                               EVAL
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNTG16,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TASTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /Z/ Z(50)                                                  Z
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IOP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      DIMENSION PZ(K), DPZIDC(K, K)                                     BNDRY
      DATA SPART/18*0/, DC/1.E-5/, MAP/8, 1, 2, 4, 7, 3, 5, 6, 0, 9/    BNDRY
      DATA BNDRYX/6HBNDRYC/                                             BNDRY
C                    SET INTEGRATION CONTROL FLAGS SO THAT FORCES WILL  BNDRY
C                    CALL ARCEN AND ARCIN                               BNDRY
      KPT = 1                                                           BNDRY
      NPTS = 1                                                          BNDRY
C                    INITIALIZE NUMBER OF TARGET MISSES COMPUTED SO FAR BNDRY
      L = 0                                                             BNDRY
C                    SET PAYOFF SIGN                                    BNDRY
      SGN = SIGN(1., SIG)                                               BNDRY
      DO 105 IARC = 1, NARC                                             BNDRY
C                    READ IN THE DATA FOR THIS SUBARC                   JUL19B
      CALL READMS(9, ARCDA, 42, IARC)                                   JUL19B
C                    SET INDEPENDENT VARIABLE                           JUL21
      X = IARC                                                          BNDRY
C                    STORE NUMBER OF C*S THAT HAVE INFLUENCE ON THIS ARC BNDRY
      NOCK = NOC(IARC)                                                  BNDRY
C                    COMPUTE RECORD SIZE FOR THIS ARC                   BNDRY
      NN = N*(NOCK + 1)                                                 BNDRY
C                    COMBINE UP TOTAL SOLUTION CORRESPONDING TO THE LAST BNDRY
C                    POINT OF THIS ARC AND STORE IN NOM ARRAY           BNDRY
      CALL READMS(41, S, NN, 2*IARC)                                    BNDRY
      CALL MATMLT(Z, S(1, 2), C, 18, NOCK, 1)                           BNDRY
      CALL MATADD(Z, Z, S, 18, 1)                                       BNDRY
      DO 101 I = 1, 18                                                  BNDRY
  101 NOM(I) = Z(I)                                                     BNDRY
      CALL FORCES                                                       BNDRY
C                    IS THIS THE LAST ARC                               BNDRY
      IF(IARC .EQ. NARC) GO TO 104                                      BNDRY   104
C                    NO.  IS THIS A BRANCH PROBLEM.                     BNDRY
```

```
      IF(NFARC .LT. NARC) GO TO 103                              BNDRY
C                    INTERMEDIATE POINT.                         BNDRY
  102 CALL INTRPT(DPZIDC, K)                                     BNDRY
      GO TO 105                                                  BNDRY
C                    IS THIS THE LAST ARC OF THE FIRST BRANCH    BNDRY
  103 IF(IARC .EQ. NFARC) GO TO 104                              BNDRY
C                    NO.  IS THIS THE BRANCH POINT               BNDRY
      IF(IARC .NE. NBRAN) GO TO 102                              BNDRY
C                    YES.  BRANCH POINT                          BNDRY
      CALL BRANPT(DPZIDC, K)                                     BNDRY
      GO TO 105                                                  BNDRY
C                    END POINT.                                  BNDRY
  104 CALL ENDPT(DPZIDC, K)                                      BNDRY
  105 CONTINUE                                                   BNDRY
C                    MAKE SURE NUMBER OF MISSES COMPUTED = NUMBER OF C*S. BNDRY
      IF(L .NE. K) CALL ERROR(BNDRYX, -1, 1)                     BNDRY
C                    MOVE MISSES TO OUTPUT ARRAY                 BNDRY
      DO 106 I = 1, K                                            BNDRY
  106 PZ(I) = PZI(I)                                             BNDRY
      RETURN                                                     BNDRY
      END                                                        BNDRY
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ARCDA	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
						BNDRY	I	ARCDA
						CHECK	I	ARCDA
						FETCH	I	ARCDA
						SALVE	I	ARCDA
						STATEF	I	SREF
						UT	I	SREF
						WRAPUP	I	ARCDA
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D /(	11)	BNDRY	I	C
						BRANPT	I	C
						GROPE	I	C
						INTRPT	I	C
						NEWCS	M	C
						NLDRV	I	C
						NOMNAL	I	C
						WRAPUP	I	C
DC	Δc_i	D	Small perturbation of a c.	/EVAL /(	867)	BNDRY	D	DC
						BRANPT	I	DC
						ENDPT	I	DC
						INTRPT	I	DC
IARC	I	M	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						OLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
KPT		D	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND	O	KPT
						BNDRY	O	KPT
						FORCES	I	KPT
						MAGIC	O	KPT
						RKUTT1	I	KPT
						SALVE	M	KPT
						WRAPUP	M	KPT
L		M	Total number of target conditions to satisfy in the problem.	/EVAL /(	868)	BNDRY	M	L
						BRANPT	M	L
						ENDPT	M	L
						INTRPT	M	L
MAP		D	A 10 word array that maps the steepest descent state vector into the OL state vector.	/EVAL /(	20)	BNDRY	D	MAP
						BRANPT	I	MAP
						ENDPT	I	MAP
						INTRPT	I	MAP
N		I	Total number of OL state and costate variables. N = 18.	/PC /(	2)	BNDRY	I	N
						CHECK	I	N
						INARC	I	N
						LINDRV	I	N
						NLDRV	I	N
						NOMNAL	I	N
						RKUTT1	I	N
						SALVE	I	N
						WRAPUP	I	N

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRO	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						OLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY	I	NBRAN
						BRANPT	I	NBRAN
						COSTAB	I	NBRAN
						ENVPRO	I	NBRAN
						INTRPT	I	NBRAN
						MAGIC	I	NBRAN
						OLTOSZ	I	NBRAN
						SALVE	I	NBRAN
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRO	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						OLTOSZ	I	NFARC
						SALVE	I	NFARC
NN		M	The number of quantities currently being numerically integrated.	/CNTRL /(	52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NOMNAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAO	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOCK	m_I	M	The number of c's in the vector C_I defined by Equation 17.4-4 of Vol.I of this document.	/EVAL /(	70)	BNDRY	M	NOCK
						BRANPT	I	NOCK
						ENDPT	I	NOCK
						INTRPT	I	NOCK
NOM	V	O	Relative velocity. (FT/SEC)	/D /(	91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NPTS		O	The total number of points in the subarc.	/CNTRL /(	19)	BCOND BNDRY FORCES INARC MAGIC SALVE WRAPUP	O O I M O M O	NPTS NPTS NPTS NPTS NPTS NPTS NPTS
PZI		I	A 40 word array that contains the target condition misses for all the target conditions in the problem.	/EVAL /(	30)	BNDRY BRANPT ENDPT INTRPT	I M M M	PZI PZI PZI PZI
S		I	An 18x41 array used to store the particular and homogeneous solutions on the early side of a corner point.	/EVAL /(	71)	BNDRY BRANPT ENDPT INTRPT	I I I I	S S S S
SGN		O	Sign of the variable SIG in the 65th word of common block /GOBAL/. SGN = +: payoff to be maximized; SGN = -: payoff to be minimized.	/EVAL /(	1)	BNDRY ENDPT	O I	SGN SGN
SIG		I	Payoff sign. SIG < 0: Payoff to be minimized; SIG > 0: Payoff to be maximized.	/GLOBAL/(	65)	BNDRY	I	SIG
SPART		O	An 18 word array whose first nine entries receive the values of the partial derivatives wrt the state of these target conditions computed in subroutine PDBCOL.	/EVAL /(	2)	BNDRY BRANPT ENDPT INTRPT	O I I I	SPART SPART SPART SPART
X	x	O	The quasitime variable.	/D /(	1)	AL4 BNDRY ERROR FETCH FORCES INARC INTERP MADAMS RKUTT1 RKUTT2 SALVE STATEF WRAPUP	I O I O I M I M M M M I M	X X X X X X X X X X X X TT
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z /(	1)	BNDRY BRANPT ENDPT ENVPRO INTERP INTRPT LINDRV NOMNAL OUTPUT RKUTT1 RKUTT2 SALVE WRAPUP	I I I I O I I M I O M M M	Z Z Z Z Z Z Z Z Z Z Z Z Z

SUBRØUTINE BRANPT

Purpose

BRANPT evaluates the state and costate target misses at a branch point and their partials with respect to the c's.*

---

* See Sections 16.6 and 17.4 of Vol. I.

```
      SUBROUTINE BRANPT(DPZIDC, KK)                                     BRANPT
C                                                                       BRANPT
C                    THIS ROUTINE EVALUATES THE STATE AND COSTATE TARGET BRANPT
C                    MISSES AT A BRANCH PT. AND THE PARTIALS OF THOSE   BRANPT
C                    MISSES W/RESP. TO THE C*S                          BRANPT
C                                                                       BRANPT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /CNTRL/                                                    CNTRL
     *NU    ;ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBORY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBORY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /EVAL/ SGN, SPART(18), MAP(10), PZI(40), NOCK, S(18, 41),  EVAL
     *TEMP(40), DZ(18), DC, L, SI(18, 41)                               EVAL
      COMMON /Z/ Z(50)                                                  Z
      DIMENSIONDPZIDC(KK, KK), AB(27, 27), BB(27), VAL(27), ZZ(18),     BRANPT
     *ZZZ(18), TIM(18), SB(18, 41), TOM(18)                             BRANPT
      MTARG = JTAB(NBRAN)                                               BRANPT
      NTARG = LTAB(NBRAN)                                               BRANPT
      ASSIGN 115 TO LABL                                                BRANPT
      KARC = NBRAN + 1                                                  BRANPT
      LARC = NFARC + 1                                                  BRANPT
  101 MA = 0                                                            BRANPT
      IF(NTARG .LE. 0) GO TO LABL                                       JUL19B
C                    SET UP THOSE ROWS OF AB THAT RESULT FROM THE INITIAL BRANPT
C                    CONDITIONS ON THE STATE                            BRANPT
      DO 102 I = 1, 27                                                  BRANPT
      DO 102 J = 1, 27                                                  BRANPT
  102 AB(I, J) = 0                                                      BRANPT
      DO 114 I = 1, 9                                                   BRANPT
C                    FIRST ARC OF FIRST BRANCH                          BRANPT
      IF(IIC(I, KARC) - 1) 103, 104, 105                                BRANPT
C                    CONTINUOUS STATE                                   BRANPT
  103 MA = MA + 1                                                       BRANPT
      AB(MA, I + 9) = 1                                                 BRANPT
      AB(MA, I + 18) = -1                                               BRANPT
      GO TO 108                                                         BRANPT
C                    KNOWN STATE                                        BRANPT
  104 MA = MA + 1                                                       BRANPT
      AB(MA, I + 9) = 1                                                 BRANPT
      GO TO 108                                                         BRANPT
  105 IF(IIC(I, KARC) - 5) 108, 106, 107                                BRANPT
C                    KNOWN DROP WEIGHT                                  BRANPT
  106 MA = MA + 1                                                       BRANPT
      AB(MA, I + 9) = -1                                                BRANPT
      AB(MA, I + 18) = 1                                                BRANPT
      GO TO 108                                                         BRANPT
C                    SIZING DROP WEIGHT                                 BRANPT
  107 MA = MA + 1                                                       BRANPT
      WPRO = GR*(ZSAVE(7) - M)                                          BRANPT
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                  BRANPT
      AB(MA, I + 9) = -1                                                BRANPT
      AB(MA, I + 18) = 1. - DWDRP                                       BRANPT
C                    FIRST ARC OF SECOND BRANCH                         BRANPT
  108 IF(IIC(I, LARC) - 1) 109, 110, 111                                BRANPT
C                    CONTINUOUS STATE                                   BRANPT
```

```
  109 MA = MA + 1                                                       BRANPT
      AB(MA, I) = 1                                                     BRANPT
      AB(MA, I + 18) = -1                                               BRANPT
      GO TO 114                                                         BRANPT
C                     KNOWN STATE                                       BRANPT
  110 MA = MA + 1                                                       BRANPT
      AB(MA, I) = 1                                                     BRANPT
      GO TO 114                                                         BRANPT
  111 IF(IIC(I, LARC) - 5) 114, 112, 113                                BRANPT
C                     KNOWN DROP WEIGHT                                 BRANPT
  112 MA = MA + 1                                                       BRANPT
      AB(MA, I) = -1                                                    BRANPT
      AB(MA, I + 18) = 1                                                BRANPT
      GO TO 114                                                         BRANPT
  113 MA = MA + 1                                                       BRANPT
      IF(IIC(I, LARC) .LT. 7) GO TO 1131                                BRANPT
C                     WEIGHT DISTRIBUTION FROM STEM TO BRANCHES.        BRANPT
      AB(MA, I) = -1                                                    BRANPT
      AB(MA, I + 9) = -1                                                BRANPT
      AB(MA, I + 18) = 1                                                BRANPT
      GO TO 114                                                         BRANPT
C                     SIZING DROP WEIGHT                                BRANPT
 1131 WPRO = GR*(ZSAVE(7) - M)                                          BRANPT
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                  BRANPT
      AB(MA, I) = -1                                                    BRANPT
      AB(MA, I + 18) = 1. - DWDRP                                       BRANPT
  114 CONTINUE                                                          BRANPT
      GO TO LABL                                                        BRANPT
C                     ARE THERE ANY STATE TARGET CONDITIONS TO BE SATIS-BRANPT
C                     FIED AT THE END OF THE STEM                       BRANPT
C                     ARE THERE ANY COSTATE TARGETS TO BE SATISFIED     BRANPT
  115 IF(MTARG .LE. 0) GO TO 121                                        BRANPT
C                     YES.  ADD ROWS TO AB MATRIX THAT RESULT FROM STATE BRANPT
C                     TARGET CONDITIONS.  AT THE SAME TIME, EVALUATE THESE BRANPT
C                     TARGETS* MISSES AND THEIR PARTIALS W/RESP. TO C*S. BRANPT
      DO 120 I = 1, MTARG                                               BRANPT
      MA = MA + 1                                                       BRANPT
      KODE = ITC(I, NBRAN)                                              BRANPT
      K = L + I                                                         BRANPT
      IF(KODE .GT. 11) GO TO 117                                        BRANPT
C                     SIMPLE RELATIVE STATE MATCH                       BRANPT
      KODE = MAP(KODE)                                                  BRANPT
      AB(MA, KODE + 18) = 1                                             BRANPT
C                     COMP. AND STORE TARGET MISS                       BRANPT
      PZI(K) = NOM(KODE) - VALTC(I, IARC)                               BRANPT
C                     STORE PARTS. W/RESP. TO C*S                       BRANPT
      DO 116 J = 1, NOCK                                                BRANPT
  116 DPZIDC(K, J) = S(KODE, J + 1)                                     BRANPT
      GO TO 120                                                         BRANPT
C                     COMPLEX CONDITION ON THE STATES                   BRANPT
  117 ISKIP = 0                                                         BRANPT
      CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                    BRANPT
      DO 118 J = 1, 9                                                   BRANPT
  118 AB(MA, J + 18) = SPART(J)                                         BRANPT
C                     COMP. AND STORE TARGET MISS.                      BRANPT
      PZI(K) = VAL - VALTC(I, NBRAN)                                    BRANPT
C                     COMP. AND STORE PARTS.W/RESP.TO C*S OF TARGET MISS BRANPT
      CALL MATMLT(TEMP, SPART, S(1, 2),1, 18, NOCK)                     BRANPT
      DO 119 J = 1, NOCK                                                BRANPT
  119 DPZIDC(K, J) = TEMP(J)                                            BRANPT
  120 CONTINUE                                                          BRANPT
C                     ARE THERE ANY COSTATE TARGET CONDITIONS           BRANPT
  121 IF(NTARG .LE.0) GO TO 141                                         BRANPT
C                     YES.  COMPLETE AND INVERT AB MATRIX               BRANPT
      CALL BASIS(AB, MA, 27)                                            BRANPT
      CALL GJRV(AB, 27, 1.E-12, IERR)                                   BRANPT
C                     SET UP THE BB VECTOR SO THAT THE TRANSVERSALITY   BRANPT
C                     CONDITIONS CAN BE EVALUATED                       BRANPT
      KNOCK = NOC(KARC)                                                 BRANPT
      KROCK = NOC(LARC)                                                 BRANPT
```

```
      CALL READMS(41, SI, 18*(KNOCK + 1), 2*NBRAN + 1)                    BRANPT
      CALL MATMLT(ZZ, SI(1, 2), C, 18, KNOCK, 1)                          BRANPT
      CALL MATADD(ZZ, ZZ, SI, 18, 1)                                      BRANPT
      CALL READMS(41, SB, 18*(KROCK + 1), 2*NFARC + 1)                    BRANPT
      CALL MATMLT(ZZZ, SB(1, 2), C, 18, KROCK, 1)                         BRANPT
      CALL MATADD(ZZZ, ZZZ, SB, 18, 1)                                    BRANPT
      DO 122 I = 1, 9                                                     BRANPT
      J = I + 9                                                           BRANPT
      BB(I) = -ZZZ(J)                                                     BRANPT
      BB(J) = -ZZ(J)                                                      BRANPT
  122 BB(J + 9) = NOM(J)                                                  BRANPT
      IF = MA + 1                                                         BRANPT
      IL = 27 - MA                                                        BRANPT
      IP = L + MTARG                                                      BRANPT
C                    COMP. THE TRANSVERSALITY CONDITIONS                  BRANPT
      CALL MATMLT(VAL, BB, AB(1, IF), 1, 27, IL)                          BRANPT
C                    STORE THE NON-TRIVIAL TRANSVERSALITY CONDITIONS AS   BRANPT
C                    COSTATE TARGET MISSES                                BRANPT
      DO 123 I = 1, NTARG                                                 BRANPT
      J = ITCT(I, NBRAN)                                                  BRANPT
      K = IP + I                                                          BRANPT
  123 PZI(K) = VAL(J)                                                     BRANPT
C                    COMPUTE THE PARTS.W/RESP.TO THE C*S OF THE COSTATE   BRANPT
C                    TARGET MISSES                                        BRANPT
      DO 140 IC = 1, KROCK                                                BRANPT
C                    COMPUTE THE STATE/COSTATE PERT. RESULTING FROM PERT. BRANPT
C                    OF THIS C.                                           BRANPT
      CALL MATMLT(DZ, SB(1, IC + 1), DC, 18, 1, 1)                        BRANPT
C                    ADD STATE/COSTATE PERT. TO BASE VALUE OF STATE/CO-   BRANPT
C                    STATE VECTOR                                         BRANPT
      CALL MATADD(TIM, ZZZ, DZ, 18, 1)                                    BRANPT
      IF(IC - KNOCK - 1) 124, 125, 127                                    BRANPT
C                    THIS C HAS AN EFFECT ON THE FIRST BRANCH.  COMP. ITS BRANPT
C                    EFFECT ON THE STATE/COSTATE VECTOR AT THE START OF   BRANPT
C                    THE FIRST BRANCH.                                    BRANPT
  124 CALL MATMLT(DZ, SI(1, IC + 1), DC, 18, 1, 1)                        BRANPT
      CALL MATADD(TOM, ZZ, DZ, 18, 1)                                     BRANPT
      GO TO 127                                                           BRANPT
C                    THIS C AND ALL SUBSEQUENT C*S IN THIS LOOP ARE TO    BRANPT
C                    HAVE NO EFFECT ON THE FIRST BRANCH.  RESTORE THE     BRANPT
C                    STATE/COSTATE VECTOR AT THE START OF THE FIRST       BRANPT
C                    BRANCH TO ITS BASE VALUE.                            BRANPT
  125 DO 126 I = 1, 18                                                    BRANPT
  126 TOM(I) = ZZ(I)                                                      BRANPT
  127 IF(IC - NOCK - 1) 128, 129, 132                                     BRANPT
C                    THIS C HAS AN EFFECT ON THE STEM. COMP. ITS EFFECT   BRANPT
C                    ON THE STATE/COSTATE VECTOR AT THE END OF THE STEM   BRANPT
  128 CALL MATMLT(DZ, S(1, IC + 1), DC, 18, 1, 1)                         BRANPT
      CALL MATADD(NOM, Z, DZ, 18, 1)                                      BRANPT
      GO TO 131                                                           BRANPT
C                    THIS C AND ALL SUBSEQUENT C*S IN THIS LOOP ARE TO    BRANPT
C                    HAVE NO EFFECT  ON THE STEM.  RESTORE THE STATE/CO-  BRANPT
C                    STATE VECTOR AT THE END OF THE STEM TO ITS BASE VAL. BRANPT
  129 DO 130 I = 1, 18                                                    BRANPT
  130 NOM(I) = Z(I)                                                       BRANPT
  131 CALL FORCES                                                         BRANPT
C                    RECOMPUTE BB VECTOR                                  BRANPT
  132 DO 133 I = 1, 9                                                     BRANPT
      J = I + 9                                                           BRANPT
      BB(I) = -TIM(J)                                                     BRANPT
      BB(J) = -TOM(J)                                                     BRANPT
  133 BB(J + 9) = NOM(7)                                                  BRANPT
C                    RECOMPUTE AB MATRIX*S INVERSE                        BRANPT
      ASSIGN 134 TO LABL                                                  BRANPT
      GO TO 101                                                           BRANPT
  134 IF(MTARG .LE. 0) GO TO 138                                          BRANPT
      DO 137 I = 1, MTARG                                                 BRANPT
      MA = MA + 1                                                         BRANPT
      KODE = ITC(I, NBRAN)                                                BRANPT
      K = L + I                                                           BRANPT
      IF(KODE .GT. 11) GO TO 135                                          BRANPT
      KODE = MAP(KODE)                                                    BRANPT
```

```
      AB(MA, KODE + 18) = 1                                           BRANPT
      GO TO 137                                                       BRANPT
  135 ISKIP = 0                                                       BRANPT
      CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                  BRANPT
      DO 136 J = 1, 9                                                 BRANPT
  136 AB(MA, J + 18) = SPART(J)                                       BRANPT
  137 CONTINUE                                                        BRANPT
  138 CALL BASIS(AB, MA, 27)                                          BRANPT
      CALL GJRV(AB, 27, 1.E-12, IERR)                                 BRANPT
C                 COMP. PERT. TRANS. CONDS.                           BRANPT
      CALL MATMLT(VAL, BB, AB(1, IF), 1, 18, IL)                      BRANPT
C                 COMP. DIVIDED DIFF. PARTIALS OF THE COSTATE         BRANPT
C                 TARGETS W/RESP. TO THIS C.                          BRANPT
      DO 139 I = 1, NTARG                                             BRANPT
      J = ITCT(I, NBRAN)                                              BRANPT
      K = IP + I                                                      BRANPT
  139 DPZIDC(K, IC) = (VAL(I) - PZI(K))/DC                            BRANPT
  140 CONTINUE                                                        BRANPT
C                 INCREMENT THE NUMBER OF STATE/COSTATE TARGET MISSES BRANPT
C                 EVALUATED SO FAR                                    BRANPT
  141 L = L + MTARG + NTARG                                           BRANPT
      RETURN                                                          BRANPT
      END                                                             BRANPT
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D /(	11)	BNDRY BRANPT GROPE INTRPT NEWCS NLDRV NOMNAL WRAPUP	I I I I M I I I	C C C C C C C C
DC	Δc_i	I	Small perturbation of a c.	/EVAL /(	867)	BNDRY BRANPT ENDPT INTRPT	D I I I	DC DC DC DC
DZ	$\Delta c_i h_i(I^-)$	I	An 18 word array that contains the second term on the right hand side of Equation 17.4-11 of Vol.I of this document.	/EVAL /(	849)	BRANPT ENDPT INTRPT	I I I	DZ DZ DZ
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5 APPLY BRANPT COSTAB COSTAI INTRPT OUTPUT PDBCOL QLTOSZ SALVE STATEF TH3	I I I I I I I I I I I I	GR GR GR GR GR GR GR GR GR GR GR GR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN BCOND BNDRY BRANPT CHECK COSTAB COSTAI ENDPT FORCES INARC INTRPT MAGIC MARCH QLTOSZ SALVE WRAPUP	I M M I M I I I I M I M I I M M	IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND BRANPT CHECK COSTAB COSTAI COSTAO INTRPT SALVE	M I I I I I I I	IIC IIC IIC IIC IIC IIC IIC IIC
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT	O I I I I I I	ITC ITC ITC ITC ITC ITC ITC
ITCT		I	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector Θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND BRANPT CHECK COSTAB COSTAI INTRPT MAGIC	O I I O O I O	ITCT ITCT ITCT ITCT ITCT ITCT ITCT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB
L		M	Total number of target conditions to satisfy in the problem.	/EVAL /(	868)	BNDRY	M	L
						BRANPT	M	L
						ENDPT	M	L
						INTRPT	M	L
LTAB		I	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND	O	LTAB
						BRANPT	I	LTAB
						COSTAB	O	LTAB
						COSTAI	O	LTAB
						INTRPT	I	LTAB
						MAGIC	M	LTAB
M	m	I	Mass (G'S)	/D /(	97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						ALDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
MAP		I	A 10 word array that maps the steepest descent state vector into the QL state vector.	/EVAL /(	20)	BNDRY	D	MAP
						BRANPT	I	MAP
						ENDPT	I	MAP
						INTRPT	I	MAP
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY	I	NBRAN
						BRANPT	I	NBRAN
						COSTAB	I	NBRAN
						ENVPRQ	I	NBRAN
						INTRPT	I	NBRAN
						MAGIC	I	NBRAN
						QLTOSZ	I	NBRAN
						SALVE	I	NBRAN
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRQ	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						QLTOSZ	I	NFARC
						SALVE	I	NFARC
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAO	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOCK	m_I	I	The number of c's in the vector C_I defined by Equation 17.4-4 of Vol.I of this document.	/EVAL /(	70)	BNDRY	M	NOCK
						BRANPT	I	NOCK
						ENDPT	I	NOCK
						INTRPT	I	NOCK

1014

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NOM	V	M	Relative velocity. (FT/SEC)	/D	/(91)	AL1 AL4 AL7 AL8 AL9 BCOND BNDRY BRANPT CONTRL ENDPT ENVPRO FETCH INTERP INTRPT NLDRV NLDRV OUTPUT PDBCOL STATEF WRAPUP	I I I I I I O M I I I O M M O I I I I I	V V V V V NOM NOM NOM V NOM V NOM V NOM NOM V V V V V
PZI		M	A 40 word array that contains the target condition misses for all the target conditions in the problem.	/EVAL	/(30)	BNDRY BRANPT ENDPT INTRPT	I M M M	PZI PZI PZI PZI
S		I	An 18x41 array used to store the particular and homogeneous solutions on the early side of a corner point.	/EVAL	/(71)	BNDRY BRANPT ENDPT INTRPT	I I I I	S S S S
SI		I	An 18x41 array used to store the particular and homogeneous solutions on the late side of a corner point.	/EVAL	/(869)	BRANPT INTRPT	I I	SI SI
SPART		I	An 18 word array whose first nine entries receive the values of the partial derivatives wrt the state of those target conditions computed in subroutine PDBCOL.	/EVAL	/(2)	BNDRY BRANPT ENDPT INTRPT	O I I I	SPART SPART SPART SPART
TEMP	$(\partial\psi_I/\partial C_I)^T$	I	A 40 word array that contains the transpose of the vector defined by Equation 17.4-9 of Vol.I of this document.	/EVAL	/(809)	BRANPT ENDPT INTRPT	I I I	TEMP TEMP TEMP
VALTC		I	A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK	/(1062)	BCOND BRANPT CHECK ENDPT INTRPT	O I I I I	VALTC VALTC VALTC VALTC VALTC
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY BRANPT ENDPT ENVPRO INTERP INTRPT LINDRV NOMNAL OUTPUT RKUTT1 RKUTT2 SALVE WRAPUP	I I I I O I I M I O M M M	Z Z Z Z Z Z Z Z Z Z Z Z Z
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D	/(151)	BCOND BRANPT COSTAB COSTAI INTRPT PDBCOL SALVE	O I I I I I I	ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE

SUBRØUTINE
CHECK

Purpose

CHECK initializes the flags and parameters needed by QL module to carry out the integration and solution of the trajectory optimization problem.

```
      PROGRAM    CHECK                                                  CHECK
C                                                                       CHECK
C                    THIS ROUTINE HANDLES THE INITIALIZATION OF THOSE   CHECK
C                    FLAGS AND PARAMETERS THAT ENABLE QL TO CARRY OUT   CHECK
C                    THE INTEGRATION AND SOLUTION OF THE OPTIMIZATION   CHECK
C                    PROBLEM.                                           CHECK
C                                                                       CHECK
      COMMON/ ORBIT/  VI,     GAMI,     PSII,     XMUI,        P,       ORBIT
     *     ECC,     AINCL,     ARGP,   ASCNOD,   SMIMAJ,   APOGEE,      ORBIT
     *  PERGEE,    ANOMLY,     CAPX,     CAPY,    ASYMP,   ENERGY,      ORBIT
     *   HMNTM,     DVIDV,    DVIDG,                                    ORBIT
     *  DVIDH ,     DVIDM,   DVIDPS,   DVIDRO,   DVIDMU,    DGIDV,      ORBIT
     *  DGIDG ,     DGIDH,   DGIDM ,   DGIDPS,   DGIDRO,    DGIDMU,     ORBIT
     *  DPIDV ,     DPIDG,   DPIDH ,   DPIDM ,   DPIDPS,    DPIDRO,     ORBIT
     *  DPIDMU,     DMIDV,   DMIDG ,   DMIDH ,   DMIDM ,    DMIDPS,     ORBIT
     *  DMIDRO,     DMIDMU,  DPDV  ,   DPDG  ,   DPDH  ,    DPDM  ,     ORBIT
     *  DPDPS ,     DPDRO ,  DPDMU ,   DECDV ,   DECDG ,    DECDH ,     ORBIT
     *  DECDM ,     DECDPS,  DECDRO,   DECDMU,   DIDV  ,    DIDG  ,     ORBIT
     *  DIDH  ,     DIDM  ,  DIDPS ,   DIDRO ,   DIDMU ,    DBEDV ,     ORBIT
     *  DBEDG ,     DBEDH ,  DBEDM ,   DBEDPS,   DBEDRO,    DBEDMU,     ORBIT
     *  DNODV ,     DNODG ,  DNODH ,   DNODM ,   DNODPS,    DNODRO,     ORBIT
     *  DNODMU,     DSMDV,    DSMDG,    DSMDH,    DSMDM,   DSMDPS       ORBIT
      COMMON/ORBIT/                                                     ORBIT
     *  DSMDRO,   DSMDMU,     DAPDV,    DAPDG,    DAPDH,   DAPDM ,      ORBIT
     *  DAPDPS,   DAPDRO,     DAPDMU,   DPEDV,    DPEDG,   DPEDH ,      ORBIT
     *DPEDM    ,DPEDPS    ,DPEDRO    ,DPEDMU    ,DANDV    ,DANDG     ,  ORBIT
     *  DANDH ,   DANDM ,     DANDPS,   DANDRO,   DANDMU,  DCXDV ,      ORBIT
     *  DCXDG ,   DCXDH ,     DCXDM ,   DCXDPS,   DCXDRO,  DCXDMU,      ORBIT
     *  DCYDV ,   DCYDG ,     DCYDH ,   DCYDM ,   DCYDPS,  DCYDRO,      ORBIT
     *  DCYDMU,   DASDV ,     DASDG ,   DASDH ,   DASDM ,  DASDPS,      ORBIT
     *  DASDRO,   DASDMU,     DENDV ,   DENDG ,   DENDH ,  DENDM ,      ORBIT
     *  DENDPS,   DENDRO,     DENDMU,   DMODV ,   DMODG ,  DMODH ,      ORBIT
     *  DMODM ,   DMODPS,     DMODRO,   DMODMU                          ORBIT
      DIMENSION ORBPRM(18),PPO(7,18)                                    ORBIT
      EQUIVALENCE (VI,ORBPRM),(DVIDV,PPO)                               ORBIT
      COMMON/ORBIT/ YMXRF,SNXLMR,CSXLMR,SDOWN,SCROSS,TD,TC              ORBIT
     *,SNPSR     ,CSPSR     ,SNGI      ,CSGI      ,SPSII     ,CPSII     , ORBIT
     * STOT      ,CSI       ,SNI       ,SNGNU     ,CSANO     ,COSDMU    , ORBIT
     * SINDMU    ,THT       ,WTFUEL                                     ORBIT
C                                                                       CHECK
      REAL MAG                                                          CHECK
      LOGICAL RECYCL                                                    CHECK
      EXTERNAL NUCASE, EXIT                                             CHECK
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,OTNC      ,DTPI      , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      COMMON /MAP/ MAP(10)                                              MAP
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), OT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
```

1018

```
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),    BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),              BLOCK
     *VALTC(10, 20), IPAY                                                 BLOCK
      DIMENSION TABLE(1), TEMP(25), ITEMP(50), NU(1)                      CHECK
      EQUIVALENCE (ITABLE, TABLE)                                         CHECK
      DATA MAP/2*5, 7, 4, 2*6, 2, 6, 4, 0/                                JUL21
      DATA RAD/57.2957795130823/                                          CHECK
      DATA CHECKX /6H CHECK/                                              CHECK
    4 FORMAT(1H1, 29HMAXIMUM NUMBER OF ITERATIONS=, I3, 13X, 17HDESIRED   CHECK
     *ACCURACY=, E13.6)                                                   CHECK
    6 FORMAT(1H , 26HADAMS-MOULTON INNER LOOPS=, I2)                      CHECK
   10 FORMAT(1H0, 23X, 27H****BOUNDARY CONDITIONS****)                    CHECK
   11 FORMAT(1H0, 10HSUBARC NO., I3/1H0, 2X, 7HPERTURB, 11X, 15HTO MATCH CHECK
     * BOUND-/1H , 12HVARIABLE NO., 7X, 17HARY CONDITION NO., 7X, 12HTO   CHECK
     * THE VALUE, 7X, 12HAT THE POINT/)                                   CHECK
   12 FORMAT(1H , 5X, I2, 11X, I8, 14X, E13.6, 11X, I3)                   CHECK
   13 FORMAT(1H0, 45HTOTAL NUMBER OF BOUNDARY CONDITIONS TO MATCH=, I3)   CHECK
      INARK = 11                                                          CHECK
C                                                                         CHECK
C                                                                         CHECK
C     ZERO OUT CONTROL BLOCK.                                             CHECK
C                                                                         CHECK
      DO 101 I = 1, 51                                                    CHECK
  101 NU(I) = 0                                                           CHECK
C                INITIALIZE REF. LATITUDE AND LONGITUDE QUANTITIES.       CHECK
      YMXRF = YMURF/RAD                                                   CHECK
      SNXLMR = SIN(XLAMRF/RAD)                                            CHECK
      CSXLMR = COS(XLAMRF/RAD)                                            CHECK
      SNPSR = SIN(PSIRF/RAD)                                              CHECK
      CSPSR = COS(PSIRF/RAD)                                              CHECK
      IERR = 0                                                            CHECK
C                                                                         CHECK
C     GET THE START TIME FOR THE CASE.                                    CHECK
C                                                                         CHECK
      CALL SECOND(TRSTR)                                                  CHECK
C                                                                         CHECK
C     CHECK AND PRINT MAXIMUM NUMBER OF ITERATIONS AND DESIRED ACCURACY.  CHECK
C                                                                         CHECK
      IF(ITRMAX .LT. 1) ITRMAX = 10                                       CHECK
      IF(ITRMAX .GT. 25) ITRMAX = 25                                      CHECK
      IF(EPSLON .LT. 1.E-8) EPSLON = 5.E-2                                CHECK
      WRITE(6, 4) ITRMAX, EPSLON                                          CHECK
C                                                                         CHECK
C     CHECK AND PRINT NUMBER OF ADAMS-MOULTON INNER LOOPS.                CHECK
C                                                                         CHECK
      IF(INNER .LT. 1) INNER = 1                                          CHECK
      IF(INNER .GT. 5) INNER = 5                                          CHECK
      WRITE(6, 6) INNER                                                   CHECK
C                                                                         CHECK
C     CHECK AND PRINT BOUNDARY CONDITIONS.                                CHECK
C                                                                         CHECK
      WRITE(6, 10)                                                        CHECK
      CALL BCOND                                                          CHECK
      WRITE(6, 14)                                                        CHECK
   14 FORMAT(1H0, 24HSTATE INITIAL CONDITIONS/)                           CHECK
   15 FORMAT(1H , 20I6)                                                   CHECK
      DO 102 I = 1, 9                                                     CHECK
  102 WRITE(6, 15) (IIC(I, J), J = 1, NARC)                               CHECK
      WRITE(6, 16)                                                        CHECK
   16 FORMAT(1H0, 26HCOSTATE INITIAL CONDITIONS/)                         CHECK
      DO 103 I = 1, 9                                                     CHECK
  103 WRITE(6, 15) (IICT(I, J), J = 1, NARC)                              CHECK
      WRITE(6, 17)                                                        CHECK
   17 FORMAT(1H0, 23HSTATE TARGET CONDITIONS/)                            CHECK
      DO 104 I = 1, 9                                                     CHECK
  104 WRITE(6, 15) (ITC(I, J), J = 1, NARC)                               CHECK
      WRITE(6, 18)                                                        CHECK
   18 FORMAT(1H0, 25HCOSTATE TARGET CONDITIONS/)                          CHECK
      DO 105 I = 1, 9                                                     CHECK
  105 WRITE(6, 15) (ITCT(I, J), J = 1, NARC)                              CHECK
C                COMPUTE MAGNITUDE OF DESIRED END CONDITION VECTOR        CHECK
      MAGBV = 0                                                           CHECK
      DO 202 IARC = 1, NARC                                               CHECK
      II = JTAB(IARC)                                                     CHECK
      IF(II .LE. 0) GO TO 202                                             CHECK   202—
```

1019

```
      DO 201 I = 1, II                                               CHECK
  201 MAGBV = MAGBV + ABS(VALTC(I, IARC))                            CHECK
  202 CONTINUE                                                       CHECK
C                   COMPUTE CONVERGENCE CRITERION FOR C SOLUTION     CHECK
      ERR = 1.E-7*SQRT(FLOAT(NOKNOW))                                CHECK
C                                                                    CHECK
C     STORE INTEGRATION STEP-SIZES.                                  CHECK
      DO 131 IARC = 1, NARC                                          CHECK
      CALL READMS(9, ARCDA, 42, IARC)                                CHECK
  131 DELT(IARC) = ARCDA(5)                                          CHECK
C                   SET MISC. FLAGS AND PARAMETERS                   CHECK
      KARD = NOKNOW                                                  CHECK
      NU = N*(KARD + 1)                                              CHECK
      NNP = N                                                        CHECK
      NUP = NNP*(KARD + 1)                                           CHECK
      LINES = 1 + (N - 1)/5                                          CHECK
      MINES = 1 + (KARD - 1)/5                                       CHECK
      KPAGE = -1                                                     CHECK
      RHOC = 1.E38                                                   CHECK
      RHOP = 1.E38                                                   CHECK
      INBDRY = .TRUE.                                                CHECK
      CALL INARC                                                     CHECK
      RETURN                                                         CHECK
      END                                                            CHECK
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ARCDA	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
						BNDRY	I	ARCDA
						CHECK	I	ARCDA
						FETCH	I	ARCDA
						SALVE	I	ARCDA
						STATEF	I	SREF
						WT	I	SREF
						WRAPUP	I	ARCDA
CSPSR	$\cos(\psi_r)$	O	Cosine of reference azimuth	/ORBIT /(	153)	CHECK	O	CSPSR
						PDBCOL	I	CSPSR
CSXLMR	$\cos(\rho-\rho_r)$	O	Cosine of refernece latitude	/ORBIT /(	147)	CHECK	O	CSXLMR
						PDBCOL	I	CSXLMR
DELT		O	A twenty word array containing the quasitime compute interval for each subarc.	/D /(	211)	CHECK	O	DELT
						ERROR	I	DELT
						INARC	I	DELT
EPSLON	ϵ	M	QL iteration convergence criterion.	/GLOBAL/(	8)	CHECK	M	EPSLON
						GROPE	I	EPSLON
ERR		O	Convergence criterion of iteration for the c's.	/D /(	8)	CHECK	O	ERR
						NEWCS	I	ERR
IARC	I	M	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND	M	IIC
						BRANPT	I	IIC
						CHECK	I	IIC
						COSTAB	I	IIC
						COSTAI	I	IIC
						COSTAO	I	IIC
						INTRPT	I	IIC
						SALVE	I	IIC
IICT		I	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK	I	IICT
						COSTAB	M	IICT
						COSTAI	M	IICT
						COSTAO	O	IICT
						MAGIC	O	IICT
						SALVE	I	IICT
INARK		O	Logical unit on which initial and converged arcs are stored. INARK = 11.	/GLOBAL/(	95)	CHECK	O	INARK
						FETCH	I	INARK
						INARC	I	INARK
						MARCH	I	INARK
						WRAPUP	I	INARK
INBDRY		O	Not used.	/CNTRL /(	30)	CHECK	O	INBDRY
INNER		M	Number of Adams-Moulton inner loops.	/GLOBAL/(	9)	CHECK	M	INNER
						MADAMS	I	INNER
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	461)	BCOND	O	ITC
						BRANPT	I	ITC
						CHECK	I	ITC
						COSTAB	I	ITC
						COSTAI	I	ITC
						ENDPT	I	ITC
						INTRPT	I	ITC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ITCT		I	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND	O	ITCT
						BRANPT	I	ITCT
						CHECK	I	ITCT
						COSTAB	O	ITCT
						COSTAI	O	ITCT
						INTRPT	I	ITCT
						MAGIC	O	ITCT
ITRMAX		M	Maximum number of QL iterations.	/GLOBAL/(	10)	CHECK	M	ITRMAX
						GROPE	I	ITRMAX
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB
KARD		M	The total number of homogeneous solutions eventually to be integrated.	/CNTRL /(	10)	CHECK	M	KARD
KPAGE		O	Not used.	/CNTRL /(	21)	CHECK	O	KPAGE
						GROPE	O	KPAGE
LINES		O	Not used.	/CNTRL /(	7)	CHECK	O	LINES
MAGBV		M	The magnitude of all of the desired values of the state target conditions.	/D /(	7)	CHECK	M	MAGBV
						NEWCS	I	MAGBV
MAP		D	An array that maps the initial arc state and costate into the QL state and costate.	/MAP /(	1)	CHECK	D	MAP
						INARC	I	MAP
MINES		O	Not used.	/CNTRL /(	20)	CHECK	O	MINES
N		I	Total number of QL state and costate variables. N = 18.	/PC /(	2)	BNDRY	I	N
						CHECK	I	N
						INARC	I	N
						LINDRV	I	N
						NLDRV	I	N
						NOMNAL	I	N
						RKUTTI	I	N
						SALVE	I	N
						WRAPUP	I	N
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRQ	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						QLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NNP		M	Number of QL state and costate variables.(18)	/CNTRL /(	22)	CHECK	M	NNP
						INTERP	I	NNP
NOKNOW		I	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK	I	NOKNOW
						COHOMO	I	NOKNOW
						COSTAB	M	NOKNOW
						COSTAI	M	NOKNOW
						COSTAO	M	NOKNOW
						GROPE	I	NOKNOW
						MAGIC	I	NOKNOW
NU		O	The largest number of quantities requiring numerical integration per QL iteration.	/CNTRL /(	1)	CHECK	O	NU
NUP		O	Same as NU.	/CNTRL /(	23)	CHECK	O	NUP
						GROPE	I	NUP
						INARC	I	NUP
PSIRF	ψ_r	I	Reference azimuth. (DEG)	/GLOBAL/(	68)	CHECK	I	PSIRF

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
RHOC		O	The magnitude of the error in the current QL iteration.	/CNTRL /(	17)	CHECK GROPE	O M	RHOC RHOC
RHOP		O	The magnitude of the error in the preceding QL iteration.	/CNTRL /(	18)	CHECK GROPE	O M	RHOP RHOP
SNPSR	$\sin(\psi_r)$	O	Sine of reference azimuth	/ORBIT /(	152)	CHECK PDBCQL	O I	SNPSR SNPSR
SNXLMR	$\sin(\rho-\rho_r)$	O	Sine of reference latitude	/ORBIT /(	146)	CHECK PDBCQL	O I	SNXLMR SNXLMR
TRSTR		I	Not used.	/CNTRL /(	25)	CHECK ETIME	I M	TRSTR TRSTR
VALTC		I	A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK /(	1062)	BCOND BRANPT CHECK ENDPT INTRPT	O I I I I	VALTC VALTC VALTC VALTC VALTC
XLAMRF	ρ_r	I	Reference latitude. (DEG)	/GLOBAL/(	4)	CHECK	I	XLAMRF
YMURF	μ_r	I	Reference longitude. (DEG)	/GLOBAL/(	5)	CHECK	I	YMURF
YMXRF	ρ_r	O	Reference longitude (RAD)	/ORBIT /(	145)	CHECK PDBCQL	O I	YMXRF YMXRF

SUBRØUTINE
CØHØMØ

Purpose

COHOMO calls in the solution for the c's.

1025


```
      PROGRAM COHOMO                                                    COHOMO
C                                                                       COHOMO
C                 THIS ROUTINE CALLS IN THE SOLUTION FOR THE C*S.       COHOMO
C                                                                       COHOMO
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      DIMENSION PARTS(40, 40)                                           COHOMO
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
C                 SET QL CONVERGENCE FLAG SO THAT FORCES DOES NOT       COHOMO
C                 WORK TOO HARD                                         COHOMO
      KONVER = .TRUE.                                                   COHOMO
      CALL NEWCS(PARTS, NOKNOW)                                         COHOMO
C                 RESTORE QL CONV. FLAG.                                COHOMO
      KONVER = .FALSE.                                                  COHOMO
      RETURN                                                            COHOMO
      END                                                               COHOMO
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
KONVER		O	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON	I	KONVER
						APPLY	I	KONVER
						ARCIN	I	KONVER
						COHOMO	O	KONVER
						GROPE	O	KONVER
						NLDRV	I	KONVER
						OUTPUT	I	KONVER
						RKUTTI	I	KONVER
NOKNOW		I	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK	I	NOKNOW
						COHOMO	I	NOKNOW
						COSTAB	M	NOKNOW
						COSTAI	M	NOKNOW
						COSTAO	M	NOKNOW
						GROPE	I	NOKNOW
						MAGIC	I	NOKNOW

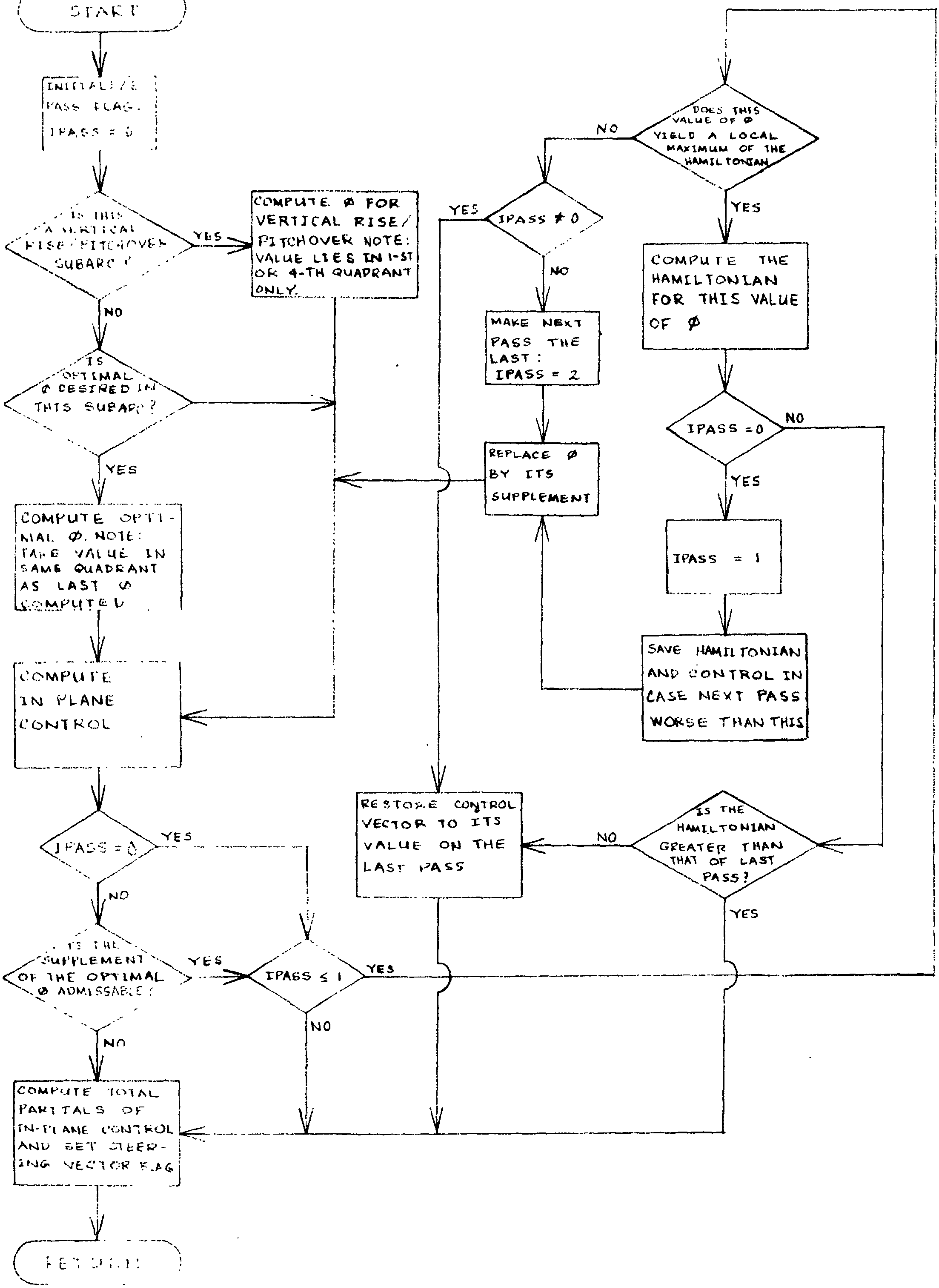
SUBRØUTINE
CØNTRL

1028

Purpose

CONTRL controls the calculation of the control vector

CNTRL


START
INITIALIZE PASS FLAG. IPASS = 0
IS THIS A VERTICAL RISE/PITCHOVER SUBARC?
YES
COMPUTE Ø FOR VERTICAL RISE/PITCHOVER NOTE: VALUE LIES IN 1-ST OR 4-TH QUADRANT ONLY.
NO
IS OPTIMAL Ø DESIRED IN THIS SUBARC?
YES
COMPUTE OPTIMAL Ø. NOTE: TAKE VALUE IN SAME QUADRANT AS LAST Ø COMPUTED
COMPUTE IN PLANE CONTROL
IPASS = 0
YES
NO
IS THE SUPPLEMENT OF THE OPTIMAL Ø ADMISSABLE?
YES
IPASS ≤ 1
YES
NO
NO
COMPUTE TOTAL PARTIALS OF IN-PLANE CONTROL AND SET STEERING VECTOR FLAG
RETURN
DOES THIS VALUE OF Ø YIELD A LOCAL MAXIMUM OF THE HAMILTONIAN
NO
YES
IPASS ≠ 0
YES
NO
MAKE NEXT PASS THE LAST: IPASS = 2
REPLACE Ø BY ITS SUPPLEMENT
COMPUTE THE HAMILTONIAN FOR THIS VALUE OF Ø
IPASS = 0
NO
YES
IPASS = 1
SAVE HAMILTONIAN AND CONTROL IN CASE NEXT PASS WORSE THAN THIS
RESTORE CONTROL VECTOR TO ITS VALUE ON THE LAST PASS
NO
IS THE HAMILTONIAN GREATER THAN THAT OF LAST PASS?
YES

```
      SUBROUTINE CONTRL                                                 CONTRL
C                                                                       CONTRL
C                    THIS ROUTINE TAKES CARE OF THE CONTROL CALCULATION CONTRL
C                                                                       CONTRL
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF       ,EJ         ,XISP       ,TAULT      ,DTNC       ,DTPI       , ARCDAT
     *IATM       ,IMODE      ,JAER       ,JPRO       ,QMAX       ,GMAX       , ARCDAT
     *XLMAX      ,HDMAX      ,GMDOT      ,ALFMAX     ,PHMAX      ,MAEA       , ARCDAT
     *MAEB       ,MAEC       ,MAED       ,MAEE       ,MAEF       ,MAEG       , ARCDAT
     *MT         ,MISP       ,MXCG       ,MZCG       ,MWDA       ,MWDB       , ARCDAT
     *MDB        ,XCGR       ,ZCGR       ,XE         ,ZE         ,XT         , ARCDAT
     *DREF       ,MCND       ,RHOB       ,QMULT      ,REMAX      ,FRATE        ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1Q  ,XK2Q  ,XK3Q  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1QT ,XK2QT ,XK3QT ,   MATS
     *XK1QD ,XK2QD ,XK3QD ,XK1QA ,XK2QA ,XK3QA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
```

1031

```
      *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,     MATS
      *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,     MATS
      *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,     MATS
      *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,     MATS
      *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,     MATS
      *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,     MATS
      *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,     MATS
      *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
      *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
      *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
      *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
      *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
      *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
      *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                MATS
       COMMON /MATS/                                                       MATS
      *OPDY(3, 8), DEPDEY(2, 8), DPDLT3, 3), PROD5(3, 64), PROD9(2, 24)    MATS--
       COMMON /MATS/                                                       MATS
      *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,     MATS
      *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,     MATS
      *POR   ,POO   ,PLG   ,PLP                                            MATS
       EQUIVALENCE(PROD1,PROD5)                                            MATS
       DATA PI/3.14159265358979/                                           CONTRL
C                          INITIALIZE PASS FLAG                            CONTRL
       IPASS = 0                                                           CONTRL
C                          IS THIS ANY NONOPTIMAL CONTROL MODE OTHER THAN VERT. CONTRL
C                          RISE/PITCHOVER.                                 CONTRL
       IF(IMODE .GT. 3) GO TO 103                                          CONTRL
C                          NO.  IS BANK ANGLE TO BE OPTIMIZED.             CONTRL
       IF(IMODE .EQ. 1) GO TO 102                                          CONTRL
C                          NO.  IS THIS VERT. RISE/PITCHOVER               CONTRL
       IF(IMODE .EQ. 3) GO TO 101                                          CONTRL
C                          NO.  OPTIMAL ANGLE OF ATTACK BUT BANK ANGLE ZERO. CONTRL
       KODE = 2                                                            CONTRL
       GO TO 103                                                           CONTRL
C                          COMPUTE BANK ANGLE FOR VERT. RISE/PITCHOVER     CONTRL
   101 XKG = (V**2/R - G)*COSGAM + OMEGA*COSRHO*(2.*V*SINPSI + R*OMEGA     CONTRL
      **(COSRHO*COSGAM + SINRHO*COSPSI*SINGAM)) - V*GAMMAD                 CONTRL
       XKP = V*COSGAM*SINRHO*(V/(R*COSRHO)*COSGAM*SINPSI + 2.*OMEGA)       CONTRL
      * + OMEGA*COSRHO*(R*OMEGA*SINRHO*SINPSI - 2.*V*COSPSI*SINGAM)        CONTRL
       DENOM = SQRT(XKG**2 + XKP**2)                                       CONTRL
       SINPHI = SIGN(1., XKG)*XKP/DENOM                                    CONTRL
       COSPHI = ABS(XKG)/DENOM                                             CONTRL
       PHI = ATAN2(SINPHI, COSPHI)                                         CONTRL
       GO TO 103                                                           CONTRL
C                          COMPUTE OPTIMAL BANK ANGLE.                     CONTRL
   102 KODE = 1                                                            CONTRL
       CGLG = LGAM*COSGAM                                                  CONTRL
       DENOM2 = LPSI**2 + CGLG**2                                          CONTRL
       DENOM = SQRT(DENOM2)                                                CONTRL
       SINPHI = LPSI/DENOM                                                 CONTRL
       IF(COSPHI*CGLG .LT. 0.) SINPHI = -SINPHI                            CONTRL
       COSPHI = SIGN(1., COSPHI)*ABS(CGLG)/DENOM                           CONTRL
       PHI = ATAN2(SINPHI, COSPHI)                                         CONTRL
       PG = SINGAM*LPSI*LGAM/DENOM2                                        CONTRL
       PLG = -LPSI*COSGAM/DENOM2                                           CONTRL
       PLP = CGLG/DENOM2                                                   CONTRL
C                          COMPUTE IN-PLANE CONTROL                        CONTRL
   103 CALL NPLANE                                                         CONTRL
C                          IS THIS THE FIRST BANK ANGLE PASS.              CONTRL
       IF(IPASS .NE. 0) GO TO 104                                          CONTRL
C                          YES.  MAY WE TRY THE SUPPLEMENT OF THE BANK ANGLE. CONTRL
       IF(IMODE .NE. 1 .OR. .NOT.ILOAD .OR. PHMAX .GT. 0.) GO TO 105       CONTRL
C                          YES.  IS THIS THE LAST BANK ANGLE PASS.         CONTRL
   104 IF(IPASS .LE. 1) GO TO 106                                          CONTRL
C                          YES.  COMPUTE PARTIALS OF IN-PLANE CONTROL W/RESP. CONTRL
C                          TO STATE AND COSTATE.                           CONTRL
   105 CALL ALGCNV(J1, J2, J3)                                             CONTRL
C                          SET CONTROL FLAG IN CASE ALPHA IS SATISFYING AN IN- CONTRL
C                          EQUALITY CONSTRAINT AND RETURN                  CONTRL
       IF(IMODE .LE. 2 .AND. J3 .NE. 1) KODE = 1 + 2*KODE                  CONTRL
       RETURN                                                              CONTRL
C                          COMPUTE NEG. SECOND PARTIAL OF HAMILTONIAN W/RESP. CONTRL
C                          TO BANK ANGLE                                   CONTRL
```

```
      106 YLTOT = (LGAM*COSPHI + LPSI*SINPHI/COSGAM)*(T*SIDAE -DB*SINA+LIFT) CONTRL
C                 TEST FOR MAXIMIZING HAMILTONIAN.                          CONTRL
          IF(YLTOT .GE. 0.) GO TO 109                                       CONTRL
C                 MINIMIZING.  IS THIS THE FIRST PASS.                      CONTRL
          IF(IPASS .NE. 0) GO TO 108                                        CONTRL
C                 YES.  MAKE THE NEXT PASS THE LAST.                        CONTRL
          IPASS = 2                                                         CONTRL
C                 COMPUTE SUPPLEMENT OF BANK ANGLE                          CONTRL
      107 PHI = PHI + SIGN(1., SINPHI)*PI                                   CONTRL
          SINPHI = -SINPHI                                                  CONTRL
          COSPHI = -COSPHI                                                  CONTRL
          KODE = 1                                                          CONTRL
C                 IF THE PRECEEDING PASS BROUGHT ABOUT THROTTLING,          CONTRL
C                 TURN THROTTLING OFF                                       CONTRL
          IF(SWITCH) J1 = 2                                                 CONTRL
          GO TO 103                                                         CONTRL
C                 RESTORE SAVED QUANTITIES FROM PRECEEDING PASS             CONTRL
      108 T = TSAV                                                          CONTRL
          DELTAE = DELSAV                                                   CONTRL
          ALPHA = ALFSAV                                                    CONTRL
          PHI = PHISAV                                                      CONTRL
          SINPHI = SINSAV                                                   CONTRL
          COSPHI = COSSAV                                                   CONTRL
          J1 = J1S                                                          CONTRL
          J3 = J3S                                                          CONTRL
          GO TO 105                                                         CONTRL
C                 COMPUTE HAMILTONIAN FOR THIS PASS                         CONTRL
      109 HSTAR = V*LV*(T*CODAE - DB*COSA - DRAG) + YLTOT                   CONTRL
C                 IS THIS THE FIRST PASS                                    CONTRL
          IF(IPASS .EQ. 0) GO TO 110                                        CONTRL
C                 WAS THIS PASS BETTER THAN THE LAST.                       CONTRL
          IF(HSTAR .GT. HSTARS) GO TO 105                                   CONTRL
          GO TO 108                                                         CONTRL
C                 SAVE NECESSARY QUANTITIES IN CASE THE NEXT PASS IS        CONTRL
C                 WORST THAN THIS PASS                                      CONTRL
      110 IPASS = 1                                                         CONTRL
          HSTARS = HSTAR                                                    CONTRL
          TSAV = T                                                          CONTRL
          DELSAV = DELTAE                                                   CONTRL
          ALFSAV = ALPHA                                                    CONTRL
          PHISAV = PHI                                                      CONTRL
          SINSAV = SINPHI                                                   CONTRL
          COSSAV = COSPHI                                                   CONTRL
          J1S = J1                                                          CONTRL
          J3S = J3                                                          CONTRL
          GO TO 107                                                         CONTRL
          END                                                               CONTRL
```

1033

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	M	Angle of attack	(RAD)	/DYNA /(	79)	AERDCO	I	ALPHA
							ALGCON	M	ALPHA
							AL2	I	ALPHA
							ARCIN	M	ALPHA
							CONTRL	M	ALPHA
							ENVPRO	I	ALPHA
							MOMECO	I	ALPHA
							NPLANE	I	ALPHA
							OUTPUT	I	ALPHA
							TRAJIN	O	ALPHA
							UT	I	ALPHA
							WRAPUP	I	ALPHA
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA /(	151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA /(	10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol		/DYNA /(	4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCOL	I	COSGAM
							STATEF	M	COSGAM
COSPHI	$\cos\phi$	M	See symbol		/DYNA /(	93)	AL1	I	COSPHI
							AL4	I	COSPHI
							APPLY	I	COSPHI
							ARCIN	O	COSPHI
							CONTRL	M	COSPHI
							OUTPUT	I	COSPHI
COSPSI	$\cos\psi$	I	See symbol		/DYNA /(	95)	AL4	I	COSPSI
							AL7	I	COSPSI
							AL8	I	COSPSI
							AL9	I	COSPSI
							CONTRL	I	COSPSI
							NLDRV	I	COSPSI
							PDBCOL	I	COSPSI
							STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol		/DYNA /(	97)	AL4	I	COSRHO
							AL7	I	COSRHO
							AL8	I	COSRHO
							AL9	I	COSRHO
							CONTRL	I	COSRHO
							NLDRV	I	COSRHO
							OUTPUT	I	COSRHO
							PDBCOL	I	COSRHO
							STATEF	M	COSRHO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DB	D_b	I	Base drag (LBS)	/DYNA /(	163)	AL1	I	DB
						AL4	I	DB
						AL6	I	DB
						AL7	I	DB
						AL8	I	DB
						AL9	I	DB
						APPLY	I	DB
						CONTRL	I	DB
						NLDRV	I	DB
						OUTPUT	I	DB
						STATEF	I	DB
						TH3	I	DB
						UT	I	DB
DELTAE	δ_E	M	Engine deflection (RADS)	/DYNA /(	155)	ALGCON	M	DELTAE
						ARCIN	M	DELTAE
						CONTRL	M	DELTAE
						DL1	I	DELTAE
						OUTPUT	I	DELTAE
						TRAJIN	O	DELTAE
						UT	I	DELTAE
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA /(	69)	AL5	I	DRAG
						AL7	I	DRAG
						AL8	I	DRAG
						AL9	I	DRAG
						APPLY	I	DRAG
						CONTRL	I	DRAG
						ENVPRO	I	DRAG
						NLDRV	I	DRAG
						OUTPUT	I	DRAG
						TH3	I	DRAG
						UT	M	DRAG
G	g	I	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA /(	8)	AL4	I	G
						AL7	I	G
						AL8	I	G
						AL9	I	G
						CONTRL	I	G
						NLDRV	I	G
						STATEF	M	G
GAMMAD		I	Pitch rate (RAD/SEC)	/DYNA /(	88)	AL4	I	GAMMAD
						ARCIN	O	GAMMAD
						CONTRL	I	GAMMAD
						NLDRV	I	GAMMAD
ILOAD		I	Logical flag that is true if there is any aerodynamic load on the vehicle.	/DYNA /(	181)	ARCIN	M	ILOAD
						CONTRL	I	ILOAD
						NPLANE	I	ILOAD
						UT	I	ILOAD
IMODE		I	Control mode option flag	/ARCDAT/(	8)	ARCIN	I	IMODE
						CONTRL	I	IMODE
						NPLANE	I	IMODE
J1		M	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY	I	J1
						ARCIN	O	J1
						CONTRL	M	J1
						FORCES	I	J1
						NPLANE	I	J1
						STATEF	I	J1
						THROTL	M	J1
J2		I	Engine deflection option flag. J2 = 1: Constant engine deflection; J2 = 2: Moment balancing.	/DYNA /(	174)	ARCIN	O	J2
						CONTRL	I	J2
						NPLANE	I	J2
J3		M	Angle of attack option flag. J3 = 1: Optimal angle of attack; J3 = 2: Constant angle of attack; J3 = 3: Untrimmed lift limit; J3 = 4: Vertical rise or pitchover; J3 = 5: Unpowered total acceleration limit; J3 = 6: Gravity turn; J3 = 7: Dynamic pressure constraint; J3 = 8: Heating rate constraint; J3 = 9: Reynolds number constraint.	/DYNA /(	175)	ARCIN	O	J3
						CONTRL	M	J3
						NPLANE	M	J3
						OUTPUT	I	J3

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
KODE		M	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.	/DYNA /(	25)	APPLY	I	KODE
						ARCIN	O	KODE
						CONTRL	M	KODE
						FORCES	I	KODE
						NLDRV	I	KODE
						STATEF	I	KODE
LGAM	λ_γ	I	Relative flight path angle costate	/D /(	101)	AL1	I	LGAM
						ARCIN	I	LGAM
						CONTRL	I	LGAM
						NLDRV	I	LGAM
						OUTPUT	I	LGAM
						WRAPUP	I	LGAM
LIFT	L	I	Aerodynamic lift (LBS)	/DYNA /(	60)	AL4	I	LIFT
						AL5	I	LIFT
						AL6	I	LIFT
						APPLY	I	LIFT
						CONTRL	I	LIFT
						ENVPRO	I	LIFT
						OUTPUT	I	LIFT
						TH3	I	LIFT
						UT	O	LIFT
LPSI	λ_ψ	I	Relative azimuth angle costate	/D /(	102)	AL1	I	LPSI
						ARCIN	I	LPSI
						CONTRL	I	LPSI
						NLDRV	I	LPSI
						OUTPUT	I	LPSI
						WRAPUP	I	LPSI
LV	λ_v	I	Relative velocity costate	/D /(	100)	AL1	I	LV
						CONTRL	I	LV
						NLDRV	I	LV
						OUTPUT	I	LV
						WRAPUP	I	LV
OMEGA	ω	I	Earth rotation rate (RAD/SEC)	/DYNA /(	5)	AL4	I	OMEGA
						AL7	I	OMEGA
						CONTRL	I	OMEGA
						PDBCQL	I	OMEGA
						TRAJIN	M	OMEGA
PG	ϕ_γ	O	See symbol	/MATS /(	551)	AL4	M	PG
						APPLY	I	PG
						ARCIN	O	PG
						CONTRL	O	PG
PHI	ϕ	M	Bank angle (RAD)	/DYNA /(	80)	CONTRL	M	PHI
						OUTPUT	I	PHI
						WRAPUP	I	PHI
PHMAX		I	Belly down flag	/ARCDAT/(	17)	CONTRL	I	PHMAX
PLG	ϕ_{λ_γ}	O	See symbol	/MATS /(	570)	APPLY	I	PLG
						ARCIN	O	PLG
						CONTRL	O	PLG
PLP	ϕ_{λ_ψ}	O	See symbol	/MATS /(	571)	APPLY	I	PLP
						ARCIN	O	PLP
						CONTRL	O	PLP
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA /(	7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCQL	I	R
						QLTOSZ	I	R
						STATEF	M	R

1036

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SIDAE	$\sin(\alpha-\delta_E)$	I	See symbol	/DYNA /(	152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE
SINA	$\sin\alpha$	I	See symbol	/DYNA /(	9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINGAM	$\sin\gamma$	I	See symbol	/DYNA /(	3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCQL	I	SINGAM
						STATEF	M	SINGAM
SINPHI	$\sin\phi$	M	See symbol	/DYNA /(	92)	AL1	I	SINPHI
						AL4	I	SINPHI
						APPLY	I	SINPHI
						CONTRL	M	SINPHI
						OUTPUT	I	SINPHI
SINPSI	$\sin\psi$	I	See symbol	/DYNA /(	94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCQL	I	SINPSI
						STATEF	O	SINPSI
SINRHO	$\sin\rho$	I	See symbol	/DYNA /(	96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCQL	I	SINRHO
						STATEF	M	SINRHO
SWITCH		I	Logical flag that is true if this is the compute point at which the powered acceleration constraint commences.	/DYNA /(	184)	CONTRL	I	SWITC
						NPLANE	I	SWITC
						THROTL	O	SWITC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T	T	M	Thrust	(LBS)	/DYNA	/(42)	ALGCON	M	T
							AL1	I	T
							AL4	I	T
							AL6	I	T
							AL7	I	T
							AL8	I	T
							AL9	I	T
							APPLY	I	T
							ARCIN	O	T
							CONTRL	M	T
							DL2	I	T
							IMPULS	I	T
							OUTPUT	I	T
							TH1	I	T
							TH2	I	T
							TH3	I	T
							TH4	I	T
V	V	I	Relative velocity.	(FT/SEC)	/D	/(91)	AL1	I	V
							AL4	I	V
							AL7	I	V
							AL8	I	V
							AL9	I	V
							BCOND	I	NOM
							BNDRY	O	NOM
							BRANPT	M	NOM
							CONTRL	I	V
							ENDPT	I	NOM
							ENVPRO	I	V
							FETCH	O	NOM
							INTERP	M	V
							INTRPT	M	NOM
							NLDRV	O	NOM
							NLDRV	I	V
							OUTPUT	I	V
							PDBCOL	I	V
							STATEF	I	V
							WRAPUP	I	V
XKG	k_γ	M	Algebraic equation used in vertical rise and pitchover		/DYNA	/(101)	AL4	I	XKG
							CONTRL	M	XKG
XKP	k_ψ	M	Algebraic equation used in vertical rise and pitchover		/DYNA	/(102)	AL4	I	XKP
							CONTRL	M	XKP

1038

SUBRØUTINE
CØSTAB

Purpose

COSTAB determines the costate initial and target conditions at a branch point.*

*See Sections 16.6, 17.1 and 17.2 in Vol. I.

```
      SUBROUTINE COSTAB                                              COSTAB
C                                                                    COSTAB
C                 THIS ROUTINE DETERMINES THE COSTATE INITIAL AND    COSTAB
C                 TARGET CONDITIONS FOR A BRANCH POINT.              COSTAB
C                                                                    COSTAB
      COMMON/GLOBAL/                                                 GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIAF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,    GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                       GLOBAL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM D
     *, LHT                                                          D
      COMMON /D/                                                     D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), BT(20), NPOINT(20), DELT(20) D
      DIMENSION NOM(20)                                              D
      EQUIVALENCE (NOM, V)                                           D
      COMMON /CNTRL/                                                 CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   , CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES , CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT, CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES       CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                 CNTRL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20), BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),        BLOCK
     *VALTC(10, 20), IPAY                                            BLOCK
      DIMENSION AB(27, 27), SPART(9), MAP(10)                        COSTAB
      DATA MAP /8 , 1, 2, 4, 7, 3, 5, 6, 0, 9/, CUSTAB/6HCOSTAB/     COSTAB
      DO 101 I = 1, 27                                               COSTAB
      DO 101 J = 1, 27                                               COSTAB
  101 AB(I, J) = 0                                                   COSTAB
      NNTTC = 0                                                      COSTAB
      MA = 0                                                         COSTAB
      LARC = NFARC + 1                                               COSTAB
C                 SET UP THOSE ROWS OF THE AB MATRIX THAT RESULT FROM COSTAB
C                 THE INITIAL CONDS. ON THE STATE.                   COSTAB
      DO 113 I = 1, 9                                                COSTAB
C                 FIRST ARC OF FIRST BRANCH                          COSTAB
      IF(IIC(I, IARC) - 1) 102, 103, 104                             COSTAB
C                 CONTINUOUS STATE                                   COSTAB
  102 MA = MA + 1                                                    COSTAB
      AB(MA, I + 9) = 1                                              COSTAB
      AB(MA, I + 18) = -1                                            COSTAB
      GO TO 107                                                      COSTAB
C                 KNOWN STATE                                        COSTAB
  103 MA = MA + 1                                                    COSTAB
      AB(MA, I + 9) = 1                                              COSTAB
      GO TO 107                                                      COSTAB
  104 IF(IIC(I, IARC) - 5) 107, 105, 106                             COSTAB
C                 KNOWN DROP WEIGHT                                  COSTAB
  105 MA = MA + 1                                                    COSTAB
      AB(MA, I + 9) = -1                                             COSTAB
      AB(MA, I + 18) = 1                                             COSTAB
      GO TO 107                                                      COSTAB
C                 SIZING DROP WEIGHT                                 COSTAB
  106 MA = MA + 1                                                    COSTAB
      CALL FETCH(-NBRAN)                                             COSTAB
      WPRO = GR*(ZSAVE(7) - M)                                       COSTAB
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                               COSTAB
      AB(MA, I + 9) = -1                                             COSTAB
      AB(MA, I +18) = 1. - DWDRP                                     COSTAB
C                 FIRST ARC OF SECOND BRANCH                         COSTAB
  107 IF(IIC(I, LARC) - 1) 108, 109, 110                             COSTAB
C                 CONTINUOUS STATE                                   COSTAB
  108 MA = MA + 1                                                    COSTAB
      AB(MA, I) = 1                                                  COSTAB
      AB(MA, I + 18) = -1                                            COSTAB
      GO TO 113                                                      COSTAB
C                 KNOWN STATE                                        COSTAB
  109 MA = MA + 1                                                    COSTAB
```

```
          AB(MA, I) = 1                                                     COSTAB
          GO TO 113                                                         COSTAB
      110 IF(IIC(I, LARC) - 5) 113, 111, 112                                COSTAB
C                        KNOWN DROP WEIGHT                                  COSTAB
      111 MA = MA + 1                                                       COSTAB
          AB(MA, I) = -1                                                    COSTAB
          AB(MA, I + 18) = 1                                                COSTAB
          GO TO 113                                                         COSTAB
      112 MA = MA + 1                                                       COSTAB
          IF(IIC(I, IARC) .LT. 7) GO TO 1121                                COSTAB
C                        WEIGHT DISTRIBUTION FROM STEM TO BRANCHES          COSTAB
          AB(MA, I) = -1                                                    COSTAB
          AB(MA, I + 9) = -1                                                COSTAB
          AB(MA, I + 18) = 1                                                COSTAB
          GO TO 113                                                         COSTAB
C                        SIZING DROP WEIGHT                                 COSTAB
     1121 CALL FETCH(-NFARC)                                                COSTAB
          WPRO = GR*(ZSAVE(7) - M)                                          COSTAB
          CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                  COSTAB
          AB(MA, I) = -1                                                    COSTAB
          AB(MA, I + 18) = 1. - DWDRP                                       COSTAB
      113 CONTINUE                                                          COSTAB
C                        STORE NUMBER OF STATE TARGET CONDS. AT END OF STEM COSTAB
          MTARG = JTAB(NBRAN)                                               COSTAB
C                        ARE THERE ANY STATE TARGET CONDS.                  COSTAB
          IF(MTARG .LE. 0) GO TO 117                                        COSTAB
C                        YES.  ADD ROWS TO AB MATRIX THAT RESULT FROM THE   COSTAB
C                        TARGET CONDS. ON THE STATE.                        COSTAB
          DO 116 I = 1, MTARG                                               COSTAB
          MA = MA + 1                                                       COSTAB
          KODE = ITC(I, NBRAN)                                              COSTAB
          IF(KODE .GT. 11) GO TO 114                                        COSTAB
C                        SIMPLE RELATIVE STATE MATCH                        COSTAB
          KODE = MAP(KODE) + 18                                             COSTAB
          AB(MA, KODE) = 1                                                  COSTAB
          GO TO 116                                                         COSTAB
C                        COMPLEX COND. ON THE STATES                        COSTAB
      114 ISKIP = 0                                                         COSTAB
          CALL FETCH(-NBRAN)                                                COSTAB
          CALL PDBCOL(KODE, DUMMY, SPART, RUNNY, 1, ISKIP)                  COSTAB
          DO 115 L = 19, 27                                                 COSTAB
      115 AB(MA, L) = SPART(L - 18)                                         COSTAB
      116 CONTINUE                                                          COSTAB
C                        ARE THERE ANY TRANSVERSALITY CONDS. AT ALL         COSTAB
      117 IF(MA .GE. 27) GO TO 132                                          COSTAB
C                        YES.  COMPLETE AND INVERT AB MATRIX.               COSTAB
          CALL BASIS(AB, MA, 27)                                            COSTAB
          CALL GJRV(AB, 27, 1.E-12, IERR)                                   COSTAB
C                        MAKE SURE INVERSION WORKED                         COSTAB
          IF(IERR .NE. 0) CALL ERROR(CUSTAB, -1, 1)                         COSTAB
C                        ANALYZE TRANSVERSALITY CONDS.                      COSTAB
          L = MA + 1                                                        COSTAB
          DO 131 J = L, 27                                                  COSTAB
          DO 118 I = 1, 9                                                   COSTAB
      118 IF(ABS(AB(I, J)) .GT. 1.E-12) GO TO 123                           COSTAB
C                        TOP THIRD OF COLUMN=ZERO, SO THIS TRANS. COND. DOES COSTAB
C                        NOT INVOLVE 2ND BRANCH                             COSTAB
          DO 119 I = 10, 18                                                 COSTAB
          DIFF = ABS(AB(I, J) - AB(I + 9, J))                               COSTAB
      119 IF(DIFF .GT. 1.E-12) GO TO 121                                    COSTAB
          KK = 0                                                            COSTAB
          DO 120 I = 10, 18                                                 COSTAB
          IF(ABS(AB(I, J)) .LE. 1.E-12) GO TO 120                           COSTAB
          KK = KK + 1                                                       COSTAB
          IF(KK .GT. 1) GO TO 130                                           COSTAB
          II = I - 9                                                        COSTAB
      120 CONTINUE                                                          COSTAB
C                        CONTINUOUS COSTATE BETWEEN STEM AND 1ST BRANCH     COSTAB
          IICT(II, IARC) = 0                                                COSTAB
          GO TO 131                                                         COSTAB
```

```
  121 KK = 0                                                          COSTAB
      DO 122 I = 10, 27                                               COSTAB
      IF(KK .EQ. 0 .AND. I .GT. 18) GO TO 130                         COSTAB
      IF(ABS(AB(I, J)) .LE. 1.E-12) GO TO 122                         COSTAB
      KK = KK + 1                                                     COSTAB
      IF(KK .GT. 1) GO TO 130                                         COSTAB
      II = I - 9                                                      COSTAB
  122 CONTINUE                                                        COSTAB
C                     KNOWN COSTATE AT START OF 1ST BRANCH            COSTAB
      IICT(II, IARC) = 1                                              COSTAB
      GO TO 131                                                       COSTAB
  123 DO 124 I = 10, 18                                               COSTAB
  124 IF(ABS(AB(I, J)) .GT. 1.E-12) GO TO 130                         COSTAB
C                     MIDDLE THIRD OF COLUMN=ZERO, SO THIS TRANS. COND.  COSTAB
C                     DOES NOT INVOLVE 1ST BRANCH                     COSTAB
      DO 125 I = 1, 9                                                 COSTAB
      DIFF = ABS(AB(I, J) - AB(I + 18, J))                            COSTAB
  125 IF(DIFF .GT. 1.E-12) GO TO 127                                  COSTAB
      KK = 0                                                          COSTAB
      DO 126 I = 1, 9                                                 COSTAB
      IF(ABS(AB(I, J)) .LE. 1.E-12) GO TO 126                         COSTAB
      KK = KK + 1                                                     COSTAB
      IF(KK .GT. 1) GO TO 130                                         COSTAB
      II = I                                                          COSTAB
  126 CONTINUE                                                        COSTAB
C                     CONTINUOUS COSTATE BETWEEN STEM AND 2ND BRANCH  COSTAB
      IICT(II, LARC) = 0                                              COSTAB
      GO TO 131                                                       COSTAB
  127 KK = 0                                                          COSTAB
      DO 129 I = 1, 27                                                COSTAB
      IF(KK .EQ. 0 .AND. I .GT. 9) GO TO 130                          COSTAB
      IF(ABS(AB(I, J)) .LE. 1.E-12) GO TO 128                         COSTAB
      KK = KK + 1                                                     COSTAB
      IF(KK .GT. 1) GO TO 130                                         COSTAB
      II = I                                                          COSTAB
  128 IF(I .EQ. 9) I = 18                                             COSTAB
  129 CONTINUE                                                        COSTAB
C                     KNOWN COSTATE AT START OF SECOND BRANCH         COSTAB
      IICT(II, LARC) = 1                                              COSTAB
      GO TO 131                                                       COSTAB
C                     IF THIS IS A LTAU MATCH AND TAU FOR THE LAST ARC  COSTAB
C                     OF THE STEM WAS KNOWN PASS THIS TRANS. COND. UP  COSTAB
C                     AS TRIVIAL                                      COSTAB
  130 IF(J .EQ. 26 .AND. IIC(8, NBRAN) .EQ. 1) GO TO 131              COSTAB
      DO 1301 I = 1, 9                                                COSTAB
      DIFF = ABS(AB(I, J) - AB(I + 9, J))                             COSTAB
      IF(DIFF .GT. 1.E-12) GO TO 1303                                 COSTAB
      DIFF = ABS(AB(I, J) - AB(I + 18, J))                            COSTAB
 1301 IF(DIFF .GT. 1.E-12) GO TO 1303                                 COSTAB
      KK = 0                                                          COSTAB
      DO 1302 I = 1, 9                                                COSTAB
      IF(ABS(AB(I, J)).LE. 1.E-12) GO TO 1302                         COSTAB
      KK = KK + 1                                                     COSTAB
      IF(KK .GT. 1) GO TO 1303                                        COSTAB
      II = I                                                          COSTAB
 1302 CONTINUE                                                        COSTAB
C                     COSTATE DISTRIBUTION ACROSS BRANCH PT           COSTAB
      IICT(II, LARC) = 7                                              COSTAB
      GO TO 131                                                       COSTAB
C                     ANOTHER NON-TRIVIAL TRANS  COND.  STORE THE SHIFTED  COSTAB
C                     COL. NO. AS A POINTER TO THE COSTATE TARGET COND.  COSTAB
 1303 NNTTC = NNTTC + 1                                               COSTAB
      ITCT(NNTTC, NBRAN) = J - MA                                     COSTAB
  131 CONTINUE                                                        COSTAB
C                     STORE THE NUMBER OF COSTATE TARGETS TO BE SATISFIED  COSTAB
C                     AT THE BRANCH POINT.                            COSTAB
  132 LTAB(NBRAN) = NNTTC                                             COSTAB
C                     UPDATE THE NUMBER OF UNKNOWN STATE AND COSTATE  COSTAB
C                     INITIAL CONDITIONS                              COSTAB
      DO 133 I = 1, 9                                                 COSTAB
```

```
      IF(IABS(IIC(I, IARC) - 3) .LT. 2) NOKNOW = NOKNOW + 1              COSTAB
  133 IF(IICT(I, IARC) .EQ. 2) NOKNOW = NOKNOW + 1                       COSTAB
      NOC(IARC) = NOKNOW                                                 COSTAB
      DO 134 I = 1, 9                                                    COSTAB
      IF(IABS(IIC(I, LARC) - 3) .LT. 2) NOKNOW = NOKNOW + 1              COSTAB
  134 IF(IICT(I, LARC) .EQ. 2) NOKNOW = NOKNOW + 1                       COSTAB
      NOC(LARC) = NOKNOW                                                 COSTAB
      RETURN                                                             COSTAB
      END                                                                COSTAB
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GR	g_r	I	Gravitational acceleration at surface of the earth (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCQL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND	M	IIC
						BRANPT	I	IIC
						CHECK	I	IIC
						COSTAB	I	IIC
						COSTAI	I	IIC
						COSTAO	I	IIC
						INTRPT	I	IIC
						SALVE	I	IIC
IICT		M	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK	I	IICT
						COSTAB	M	IICT
						COSTAI	M	IICT
						COSTAO	O	IICT
						MAGIC	O	IICT
						SALVE	I	IICT
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND	O	ITC
						BRANPT	I	ITC
						CHECK	I	ITC
						COSTAB	I	ITC
						COSTAI	I	ITC
						ENDPT	I	ITC
						INTRPT	I	ITC
ITCT		O	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector Θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND	O	ITCT
						BRANPT	I	ITCT
						CHECK	I	ITCT
						COSTAB	O	ITCT
						COSTAI	O	ITCT
						INTRPT	I	ITCT
						MAGIC	O	ITCT
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LTAB		O	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND BRANPT COSTAB COSTAI INTRPT MAGIC	O I O O I M	LTAB LTAB LTAB LTAB LTAB LTAB
M	m	I	Mass (G'S)	/D /(	97)	AL4 AL7 AL8 AL9 APPLY BRANPT COSTAB COSTAI INTRPT NLORV OUTPUT SALVE STATEF WRAPUP	I I I I I I I I I I I I I I	M M M M M M M M M M M M M M
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY BRANPT COSTAB ENVPRQ INTRPT MAGIC QLTOSZ SALVE	I I I I I I I I	NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND BNDRY BRANPT COSTAB ENVPRQ INTRPT MAGIC QLTOSZ SALVE	I I I I I I I I I	NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC
NOC		O	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY BRANPT COSTAB COSTAI COSTAO INARC INTRPT SALVE WRAPUP	I I O O O I I I I	NOC NOC NOC NOC NOC NOC NOC NOC NOC
NOKNOW		M	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK COMOMO COSTAB COSTAI COSTAO GROPE MAGIC	I I M M M I I	NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW NOKNOW
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D /(	151)	BCOND BRANPT COSTAB COSTAI INTRPT PDBCOL SALVE	O I I I I I I	ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE ZSAVE

1046

SUBRØUTINE CØSTAI

1047

Purpose

COSTAI determines the costate initial and target conditions at an intermediate point.*

*See Sections 16.6, 17.1 and 17.2 in Vol. I.

```
      SUBROUTINE COSTAI                                                 COSTAI
C                                                                       COSTAI
C                     THIS ROUTINE DETERMINES THE COSTATE INITIAL CONDI-  COSTAI
C                     TIONS ON THE LATE SIDE OF AN INTERMEDIATE POINT AND COSTAI
C                     WHICH TRANSVERSALITY CONDITIONS, IF ANY, MUST BE    COSTAI
C                     SATISFIED ON THE EARLY SIDE.                        COSTAI
C                                                                       COSTAI
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LAU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      DIMENSION AI(18, 18), SPART(9), MAP(10)                           COSTAI
      DATA MAP /8,1,2,4,7,3,5,6,0,9/, CUSTAI/6HCOSTAI/                  COSTAI
C                     INITIALIZE THE NUMBER OF NON-TRIVIAL TRANS. CONDS.  COSTAI
C                     AND THE NUMBER OF STATE INITIAL AND TARGET CONDS.   COSTAI
C                     FOR THIS INTERMEDIATE PT.                           COSTAI
      NNTTC = 0                                                         COSTAI
      MA = 0                                                            COSTAI
      DO 101 I = 1, 18                                                  COSTAI
      DO 101 J = 1, 18                                                  COSTAI
  101 AI(I, J) = 0                                                      COSTAI
C                     SET UP THOSE ROWS OF THE AI MATRIX THAT RESULT FROM COSTAI
C                     STATE INITIAL CONDITIONS                            COSTAI
      DO 107 I = 1, 9                                                   COSTAI
      IF(IIC(I, IARC) - 1) 102, 103, 104                                COSTAI  102 103 104
C                     CONTINUOUS STATE                                    COSTAI
  102 MA = MA + 1                                                       COSTAI
      AI(MA, I) = 1                                                     COSTAI
      AI(MA, I + 9) = -1                                                COSTAI
      GO TO 107                                                         COSTAI  107
C                     KNOWN STATE                                         COSTAI
  103 MA = MA + 1                                                       COSTAI
      AI(MA, I) = 1                                                     COSTAI
      GO TO 107                                                         COSTAI  107
  104 IF(IIC(I, IARC) - 5) 107, 105, 106                                COSTAI  105 106 107
C                     FIXED DROP WEIGHT                                   COSTAI
  105 MA = MA + 1                                                       COSTAI
      AI(MA, I) = -1                                                    COSTAI
      AI(MA, I + 9) = 1                                                 COSTAI
      GO TO 107                                                         COSTAI  107
C                     SIZING DROP WEIGHT                                  COSTAI
  106 MA = MA + 1                                                       COSTAI
      CALL FETCH(-IARC + 1)                                             COSTAI
      WPRO = GR*(ZSAVE(7) - M)                                          COSTAI
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                  COSTAI
      AI(MA, I) = -1                                                    COSTAI
      AI(MA, I + 9) = 1. - DWDRP                                        COSTAI
  107 CONTINUE                                                          COSTAI
C                     STORE NUMBER OF STATE TARGETS                       COSTAI
      MTARG = JTAB(IARC - 1)                                            COSTAI
C                     ARE THAN ANY STATE TARGETS AT THIS POINT            COSTAI
      IF(MTARG .LE. 0) GO TO 111                                        COSTAI  111
C                     YES.  ADD THOSE ROWS TO AI RESULTING FROM TARGETS.  COSTAI
      DO 110 I = 1, MTARG                                               COSTAI
```

1049

```
      MA = MA + 1                                                  COSTAI
      KODE = ITC(I, IARC - 1)                                      COSTAI
      IF(KODE .GT. 11) GO TO 108                                   COSTAI
C                 SIMPLE RELATIVE STATE MATCH                      COSTAI
      KODE = MAP(KODE) + 9                                         COSTAI
      AI(MA, KODE) = 1                                             COSTAI
      GO TO 110                                                    COSTAI
C                 COMPLEX TARGET CONDITION ON STATES               COSTAI
  108 ISKIP = 0                                                    COSTAI
      CALL FETCH(-IARC + 1)                                        COSTAI
      CALL PDBCOL(KODE, DUMMY, SPART, RUMMY, 1, ISKIP)             COSTAI
      DO 109 L = 10, 18                                            COSTAI
  109 AI(MA, L) = SPART(L - 9)                                     COSTAI
  110 CONTINUE                                                     COSTAI
C                 ARE THERE ANY TRANSVERSALITY CONDITIONS AT THIS PT.  COSTAI
  111 IF(MA .GE. 18) GO TO 121                                     COSTAI
C                 YES.  COMPLETE AND INVERT AI MATRIX.             COSTAI
      CALL BASIS(AI, MA, 18)                                       COSTAI
      CALL GJRV(AI, 18, 1.E-12, IERR)                              COSTAI
C                 MAKE SURE INVERSION WORKED                       COSTAI
      IF(IERR .NE. 0) CALL ERROR(CUSTAI, -1, 1)                    COSTAI
C                 ANALYZE TRANSVERSALITY CONDS.                    COSTAI
      L = MA + 1                                                   COSTAI
      DO 120 J = L, 18                                             COSTAI
C                 CHECK FOR CONTINUOUS COSTATE                     COSTAI
      DO 112 I = 1, 9                                              COSTAI
      DIFF = ABS(AI(I, J) - AI(I + 9, J))                          COSTAI
      IF(DIFF .GT. 1.E-12) GO TO 115                               COSTAI
  112 CONTINUE                                                     COSTAI
C                 TOP HALF OF COL. EQUALS BOTTOM HALF.  CHECK NUMBER  COSTAI
C                 OF NONZERO ENTRIES.                              COSTAI
      KK = 0                                                       COSTAI
      DO 114 I = 1, 9                                              COSTAI
      IF(ABS(AI(I, J)).LE. 1.E-12) GO TO 114                       COSTAI
      KK = KK + 1                                                  COSTAI
      IF(KK - 1) 114, 113, 119                                     COSTAI
  113 II = I                                                       COSTAI
  114 CONTINUE                                                     COSTAI
C                 CONTINUOUS COSTATE                               COSTAI
      IICT(II, IARC) = 0                                           COSTAI
      GO TO 120                                                    COSTAI
C                 CHECK FOR KNOWN COSTATE                          COSTAI
  115 KK = 0                                                       COSTAI
      DO 118 I = 1, 18                                             COSTAI
      IF(KK .EQ. 0 .AND. I .GT. 9) GO TO 119                       COSTAI
      IF(ABS(AI(I, J)).LE. 1.E-12) GO TO 118                       COSTAI
      KK = KK + 1                                                  COSTAI
      IF(KK - 1) 118,117,119                                       COSTAI
  117 II = I                                                       COSTAI
  118 CONTINUE                                                     COSTAI
C                 KNOWN COSTATE                                    COSTAI
      IICT(II, IARC) = 1                                           COSTAI
      GO TO 120                                                    COSTAI
C                 IF THIS IS A LTAU MATCH AND TAU FOR PRECEEDING ARC  COSTAI
C                 IS KNOWN, PASS UP TRANS. COND. AS TRIVIAL.       COSTAI
  119 IF(J .EQ. 17 .AND. IIC(8, IARC - 1) .EQ. 1) GO TO 120        COSTAI
C                 ANOTHE NON-TRIV. TRANS. COND.  STORE SHIFTED COLUMN  COSTAI
C                 NO. IN COSTATE TARG. COND. ARRAY                 COSTAI
      NNTTC = NNTTC + 1                                            COSTAI
      ITCT(NNTTC, IARC - 1) = J - MA                               COSTAI
  120 CONTINUE                                                     COSTAI
C                 STORE TOTAL NUMBER OF NON-TRIV. TRANS. CONDS. FOR  COSTAI
C                 THIS POINT                                       COSTAI
  121 LTAB(IARC - 1) = NNTTC                                       COSTAI
C                 UPDATE TOTAL NUMBER OF UNKNOWN INITIAL STATES/CO-  COSTAI
C                 STATES THROUGH THIS ARC AND STORE RESULT         COSTAI
      DO 122 I = 1, 9                                              COSTAI
      IF(IABS(IIC(I, IARC) - 3) .LT. 2) NOKNOW = NOKNOW + 1        COSTAI
  122 IF(IICT(I, IARC) .EQ. 2) NOKNOW = NOKNOW + 1                 COSTAI
      NOC(IARC) = NOKNOW                                           COSTAI
```

1050

```
143.        RETURN                                                    COSTAI
144.        END                                                       COSTAI
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5 APPLY BRANPT COSTAB COSTAI INTRPT OUTPUT PDBCOL QLTOSZ SALVE STATEF TH3	I I I I I I I I I I I I	GR GR GR GR GR GR GR GR GR GR GR GR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN BCOND BNDRY BRANPT CHECK COSTAB COSTAI ENDPT FORCES INARC INTRPT MAGIC MARCH QLTOSZ SALVE WRAPUP	I M M I M I I I I M I M I I M M	IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND BRANPT CHECK COSTAB COSTAI COSTAO INTRPT SALVE	M I I I I I I I	IIC IIC IIC IIC IIC IIC IIC IIC
IICT		M	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK COSTAB COSTAI COSTAO MAGIC SALVE	I M M O O I	IICT IICT IICT IICT IICT IICT
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT	O I * I I I I	ITC ITC ITC ITC ITC ITC ITC
ITCT		O	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND BRANPT CHECK COSTAB COSTAI INTRPT MAGIC	O I I O O I O	ITCT ITCT ITCT ITCT ITCT ITCT ITCT
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT MAGIC	M I I I I I I I	JTAB JTAB JTAB JTAB JTAB JTAB JTAB JTAB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LTAB		O	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND	O	LTAB
						BRANPT	I	LTAB
						COSTAB	O	LTAB
						COSTAI	O	LTAB
						INTRPT	I	LTAB
						MAGIC	M	LTAB
M	m	I	Mass (G'S)	/D /(	97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
NOC		O	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAO	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOKNOW		M	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK	I	NOKNOW
						COHOMO	I	NOKNOW
						COSTAB	M	NOKNOW
						COSTAI	M	NOKNOW
						COSTAO	M	NOKNOW
						GROPE	I	NOKNOW
						MAGIC	I	NOKNOW
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D /(	151)	BCOND	O	ZSAVE
						BRANPT	I	ZSAVE
						COSTAB	I	ZSAVE
						COSTAI	I	ZSAVE
						INTRPT	I	ZSAVE
						PDBCOL	I	ZSAVE
						SALVE	I	ZSAVE

SUBRØUTINE
CØSTAO

1054

Purpose

COSTA0 determines the costate initial conditions at the initial point.*

*See Sections 16.6 and 17.1 in Vol. I.

```
      SUBROUTINE COSTAO                                                 COSTAO
C                                                                       COSTAO
C                    THIS ROUTINE DETERMINES THE COSTATE INITIAL CONDS. COSTAO
C                    AT THE INITIAL                                     COSTAO
C                                                                       COSTAO
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      DIMENSION AO(9, 9)                                                COSTAO
      DO 101 I = 1, 9                                                   COSTAO
      DO 101 J = 1, 9                                                   COSTAO
  101 AO(I, J) = 0                                                      COSTAO
      MA = 0                                                            COSTAO
C                    SET UP THOSE ROWS OF AO RESULTING FROM STATE INITIAL COSTAO
C                    CONDS                                              COSTAO
      DO 102 I = 1, 9                                                   COSTAO
      IF(IIC(I, 1) .NE. 1) GO TO 102                                    COSTAO
      MA = MA + 1                                                       COSTAO
      AO(MA, I) = 1                                                     COSTAO
  102 CONTINUE                                                          COSTAO
C                    ARE THERE ANY KNOWN COSTATES AT THE INITIAL PT.    COSTAO
      IF(MA .GE. 9) GO TO 105                                           COSTAO
C                    YES.  FIND OUT WHICH ONES ARE KNOWN                COSTAO
      CALL BASIS(AO, MA, 9)                                             COSTAO
      CALL GJRV(AO, 9, 1.E-12, IERR)                                    COSTAO
      L = MA + 1                                                        COSTAO
      DO 104 J = L, 9                                                   COSTAO
      DO 103 I = 1, 9                                                   COSTAO
      IF(ABS(AO(I, J)) .LE. 1.E-12) GO TO 103                           COSTAO
      IICT(I, 1) = 1                                                    COSTAO
      GO TO 104                                                         COSTAO
  103 CONTINUE                                                          COSTAO
  104 CONTINUE                                                          COSTAO
C                    SET TOTAL NUMBER OF UNKNOWN INITIAL STATES/COSTATES COSTAO
C                    TO 8 AND STORE.                                    COSTAO
  105 NOKNOW = 8                                                        COSTAO
      NOC(1) = NOKNOW                                                   COSTAO
      RETURN                                                            COSTAO
      END                                                               COSTAO
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND	M	IIC
						BRANPT	I	IIC
						CHECK	I	IIC
						COSTAB	I	IIC
						COSTAI	I	IIC
						COSTAO	I	IIC
						INTRPT	I	IIC
						SALVE	I	IIC
IICT		O	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK	I	IICT
						COSTAB	M	IICT
						COSTAI	M	IICT
						COSTAO	O	IICT
						MAGIC	O	IICT
						SALVE	I	IICT
NOC		O	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAO	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOKNOW		M	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK	I	NOKNOW
						COHOMO	I	NOKNOW
						COSTAB	M	NOKNOW
						COSTAI	M	NOKNOW
						COSTAO	M	NOKNOW
						GROPE	I	NOKNOW
						MAGIC	I	NOKNOW

SUBRØUTINE
DL1

Purpose

DLI evaluates the constant engine deflection constraint,

$$\delta_E - C_{\delta_E} = 0.$$

```
      SUBROUTINE DL1                                                    DL1
C                                                                       DL1
C                 THIS ROUTINE APPLIES WHEN ENGINE DEFLECTION IS A      DL1
C                 CONSTANT                                              DL1
C                                                                       DL1
C                                                                       DL1
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDV(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
```

1060

```
      COMMON /MATS/                                                    MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,  MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,  MATS
     *POR   ,POO   ,PLG   ,PLP                                         MATS
      EQUIVALENCE(PROD1,PROD5)                                         MATS
C                                                                      DL1
      ENTRY DL1002                                                     DL1
      ENTRY DL1001                                                     DL1
   40 XK2D = 1.                                                        DL1
      ENTRY DL1000                                                     DL1
   50 XK2 = DELTAE - CDE                                               DL1
C                                                                      DL1
      RETURN                                                           DL1
      END                                                              DL1
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CDE	C_{δ_E}	I	Value for engine deflection in case constant engine deflection constraint is used	(RADS)	/DYNA	/(156)	DL1	I	CDE
DELTAE	δ_E	I	Engine deflection	(RADS)	/DYNA	/(155)	ALGCON	M	DELTAE
							ARCIN	M	DELTAE
							CONTRL	M	DELTAE
							DL1	I	DELTAE
							OUTPUT	I	DELTAE
							TRAJIN	O	DELTAE
							UT	I	DELTAE

SUBRØUTINE
DL2

Purpose

DL2 evaluates the moment balance constraint, Equation 3.1-17in Vol. I. In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

106.1


```
      SUBROUTINE DL2                                                    DL2
C                                                                       DL2
C                  THIS ROUTINE APPLIES WHEN ENGINE DEFLECTION IS NON-  DL2
C                  CONSTANT                                             DL2
C                                                                       DL2
C                                                                       DL2
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      ,ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      ,ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      ,ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      ,ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      ,ARCDAT
     *MOB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        ,ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE      ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMAM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XI    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
```

```
      *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,       MATS
      *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,       MATS
      *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,       MATS
      *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,       MATS
      *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,       MATS
      *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,       MATS
      *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,       MATS
      *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2                  MATS
       COMMON /MATS/                                                         MATS
      *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)      MATS
       COMMON /MATS/                                                         MATS
      *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,       MATS
      *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,       MATS
      *POR   ,POO   ,PLG   ,PLP                                              MATS
       EQUIVALENCE(PROD1,PROD5)                                              MATS
C                                                                            DL2
C                     THIS ENTRY COMP. 2ND PARTS.W/RESP. TO STATE            DL2
       ENTRY DL2020                                                          DL2
       ASSIGN 6 TO IGO                                                       DL2
       GO TO 4                                                               DL2
C                     THIS ENTRY COMP. MIXED PARTS. W/RESP. TO STATE AND     DL2
C                     CONTROL.                                               DL2
       ENTRY DL2011                                                          DL2
       ASSIGN 10 TO IGO                                                      DL2
       GO TO 4                                                               DL2
C                     THIS ENTRY COMP. 1ST PARTS. W/RESP. TO STATE           DL2
       ENTRY DL2010                                                          DL2
       ASSIGN 20 TO IGO                                                      DL2
       GO TO 4                                                               DL2
C                     THIS ENTRY COMP. 2ND PARTS. W/RESP. TO CONTROL         DL2
       ENTRY DL2002                                                          DL2
       ASSIGN 30 TO IGO                                                      DL2
       GO TO 5                                                               DL2
C                     THIS ENTRY COMP. 1ST PARTS. W/RESP. TO CONTROL         DL2
       ENTRY DL2001                                                          DL2
       ASSIGN 40 TO IGO                                                      DL2
       GO TO 5                                                               DL2
C                     THIS ENTRY EVAL. THE CONSTRAINING EQ. ONLY             DL2
       ENTRY DL2000                                                          DL2
       ASSIGN 50 TO IGO                                                      DL2
       GO TO 5                                                               DL2
C                                                                            DL2
C                                                                            DL2
     4 RAC = ZCGM*COD - XCGM*SID                                             DL2
       HAC = -ZCGM*SID - XCGM*COD                                            DL2
     5 XEMXCG = XE - XCG                                                     DL2
       ZEMZCG = ZE - ZCG                                                     DL2
       FAC = XEMXCG*SID - ZEMZCG*COD                                         DL2
       SAC = XEMXCG*COD + ZEMZCG*SID                                         DL2
       XJ1 = 1.- XJ                                                          DL2
       GO TO IGO                                                             DL2
     6 XK2VV = XJVV*XMCG + 2 *XJV*XMCGV - XJ1*XMCGVV                         DL2
       XK2RV = XJVR*XMCG + XJV*XMCGR + XJR*XMCGV - XJ1*XMCGVR                DL2
       XK2MV = XJV*XMCGM - XJ1*XMCGVM                                        DL2
       XK2RR = XJRR*XMCG + 2.*XJR*XMCGR - XJ1*XMCGRR                         DL2
       XK2MR = XJR*XMCGM - XJ1*XMCGRM                                        DL2
       XK2MM = (ZCGMM*COD - XCGMM*SID)*T - XJ1*XMCGMM                        DL2
    10 XK2VA = XJV*XMCGA - XJ1*XMCGVA                                        DL2
       XK2RA = XJR*XMCGA - XJ1*XMCGRA                                        DL2
       XK2MA = -XJ1*XMCGMA                                                   DL2
       XK2MT = RAC                                                           DL2
       XK2MD = HAC*T                                                         DL2
    20 XK2V = XJV*XMCG - XJ1*XMCGV                                           DL2
       XK2R = XJR*XMCG - XJ1*XMCGR                                           DL2
       XK2M = RAC*T - XJ1*XMCGM                                              DL2
    30 XK2TD = SAC                                                           DL2
       XK2DD = -T*FAC                                                        DL2
       XK2AA = -XJ1*XMCGAA                                                   DL2
    40 XK2T = FAC                                                            DL2
       XK2D = T*(XEMXCG*COD + ZEMZCG*SID)                                    DL2
       XK2A = -XJ1*XMCGA                                                     DL2
    50 XK2 = FAC*T - XJ1*XMCG                                                DL2
C                                                                            DL2
       RETURN                                                                DL2
```

9920.1

150. END DL2

6 OCT 72 6.01-44

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
COD	$\cos\delta_E$	I	See symbol	/DYNA /(	153)	DL2 OUTPUT TH3 UT	I I I M	COD COD COD COD
SID	$\sin\delta_E$	I	See symbol	/DYNA /(	154)	DL2 OUTPUT TH3 UT	I I I M	SID SID SID SID
T	T	I	Thrust (LBS)	/DYNA /(	42)	ALGCON AL1 AL4 AL6 AL7 AL8 AL9 APPLY ARCIN CONTRL DL2 IMPULS OUTPUT TH1 TH2 TH3 TH4	M I I I I I I I O M I I I I I I I	T T T T T T T T T T T T T T T T T
XCG	X_{CG}	I	Center of gravity body x station (FT)	/DYNA /(	157)	DL2 STATEF UT	I I I	XCG XCG XCG
XCGM	$\partial X_{CG}/\partial m$	I	See symbol	/DYNA /(	108)	DL2 STATEF UT	I M I	XCGM XCGM XCGM
XCGMM	$\partial^2 X_{CG}/\partial m^2$	I	See symbol	/DYNA /(	109)	DL2 STATEF UT	I M I	XCGMM XCGMM XCGMM
XE	X_E	I	Engine thrust centroid body x station	/ARCDAT/(	34)	DL2	I	XE
XJ	$\dot{j}$	I	Control blend factor	/DYNA /(	159)	ARCIN DL2 OUTPUT STATEF UT	O I I I I	XJ XJ XJ XJ XJ
XJR	$\partial\dot{j}/\partial R$	I	See symbol	/DYNA /(	113)	DL2 STATEF UT	I O I	XJR XJR XJR
XJRR	$\partial^2\dot{j}/\partial R^2$	I	See symbol	/DYNA /(	116)	DL2 STATEF UT	I O I	XJRR XJRR XJRR
XJV	$\partial\dot{j}/\partial V$	I	See symbol	/DYNA /(	112)	DL2 STATEF UT	I O I	XJV XJV XJV
XJVR	$\partial^2\dot{j}/\partial V\partial R$	I	See symbol	/DYNA /(	115)	DL2 STATEF UT	I O I	XJVR XJVR XJVR
XJVV	$\partial^2\dot{j}/\partial V^2$	I	See symbol	/DYNA /(	114)	DL2 STATEF UT	I O I	XJVV XJVV XJVV
XMCG	M_{CG}	I	Aerodynamic moment about center of gravity (FT-LBS)	/DYNA /(	160)	DL2 OUTPUT UT	I I M	XMCG XMCG XMCG
XMCGA	$\partial M_{CG}/\partial\alpha$	I	See symbol	/DYNA /(	176)	DL2 UT	I M	XMCGA XMCGA
XMCGAA	$\partial^2 M_{CG}/\partial\alpha^2$	I	See symbol	/DYNA /(	169)	DL2 UT	I M	XMCGAA XMCGAA
XMCGM	$\partial M_{CG}/\partial m$	I	See symbol	/DYNA /(	203)	DL2 UT	I M	XMCGM XMCGM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XMCGMA	$\partial^2 M_{CG}/\partial m \partial \alpha$	I	See symbol		/DYNA /(	212)	DL2 UT	I M	XMCGMA XMCGMA
XMCGMM	$\partial^2 M_{CG}/\partial m^2$	I	See symbol		/DYNA /(	211)	DL2 UT	I M	XMCGMM XMCGMM
XMCGR	$\partial M_{CG}/\partial R$	I	See symbol		/DYNA /(	202)	DL2 UT	I M	XMCGR XMCGR
XMCGRA	$\partial^2 M_{CG}/\partial R \partial \alpha$	I	See symbol		/DYNA /(	210)	DL2 UT	I M	XMCGRA XMCGRA
XMCGRM	$\partial^2 M_{CG}/\partial R \partial m$	I	See symbol		/DYNA /(	209)	DL2 UT	I M	XMCGRM XMCGRM
XMCGRR	$\partial^2 M_{CG}/\partial \alpha^2$	I	See symbol		/DYNA /(	208)	DL2 UT	I M	XMCGRR XMCGRR
XMCGV	$\partial M_{CG}/\partial V$	I	See symbol		/DYNA /(	201)	DL2 UT	I M	XMCGV XMCGV
XMCGVA	$\partial^2 M_{CG}/\partial V \partial \alpha$	I	See symbol		/DYNA /(	207)	DL2 UT	I M	XMCGVA XMCGVA
XMCGVM	$\partial^2 M_{CG}/\partial V \partial m$	I	See symbol		/DYNA /(	206)	DL2 UT	I M	XMCGVM XMCGVM
XMCGVR	$\partial^2 M_{CG}/\partial V \partial R$	I	See symbol		/DYNA /(	205)	DL2 UT	I M	XMCGVR XMCGVR
XMCGVV	$\partial^2 M_{CG}/\partial V^2$	I	See symbol		/DYNA /(	204)	DL2 UT	I M	XMCGVV XMCGVV
ZCG	$-Z_{CG}$	I	Center of gravity body z station	(FT)	/DYNA /(	158)	DL2 STATEF UT	I I I	ZCG ZCG ZCG
ZCGM	$\partial Z_{CG}/\partial m$	I	See symbol		/DYNA /(	110)	DL2 STATEF UT	I M I	ZCGM ZCGM ZCGM
ZCGMM	$\partial^2 Z_{CG}/\partial m^2$	I	See symbol		/DYNA /(	111)	DL2 STATEF UT	I M I	ZCGMM ZCGMM ZCGMM
ZE	Z_E	I	Engine thrust centroid body z station		/ARCDAT/(	35)	DL2 UT	I I	ZE ZE

SUBRØUTINE
ENDPT

1070

Purpose

ENDPT evaluates the state and costate target misses at the endpoint of a branch. It also computes the partials of these misses with respect to the c's*.

---

*See Sections 16.6 and 17.4 of Vol. I.

ENDPT

```
      SUBROUTINE ENDPT(DPZIDC, KK)                                     ENDPT
C                                                                      ENDPT
C                   THIS ROUTINE EVALUATES THE STATE AND COSTATE TARGET  ENDPT
C                   MISSES AT THE END POINT OF A BRANCH.  IT ALSO COM-   ENDPT
C                   PUTES THE PARTIALS OF THOSE MISSES WITH RESPECT TO   ENDPT
C                   THE C*S.                                           ENDPT
C                                                                      ENDPT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM  D
     *, LHT                                                            D
      COMMON /D/                                                       D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,  JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20) D
      DIMENSION NOM(20)                                                D
      EQUIVALENCE (NOM, V)                                             D
      COMMON /CNTRL/                                                   CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,  CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNTC16,RHOC  ,RHOP  ,NPTS  ,MINES ,  CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,  CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES         CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                   CNTRL
      COMMON/GLOBAL/                                                   GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,  GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIAF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,      GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                         GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20), BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),          BLOCK
     *VALTC(10, 20), IPAY                                              BLOCK
      COMMON /EVAL/ SGN, SPART(18), MAP(10), PZI(40), NOCK, S(18, 41), EVAL
     *TEMP(40), DZ(18), DC, L, SI(18, 41)                              EVAL
      COMMON /Z/ Z(50)                                                 Z
      DIMENSION AF(9, 9), BF(9), DPZIDC(KK, KK), VAL(9)                ENDPT
      DATA ENDPX/6HENDPT /                                             ENDPT
C                   MOVE COSTATE INTO BF VECTOR.                       ENDPT
      DO 101 I = 1, 9                                                  ENDPT
      BF(I) = NOM(I + 9)                                               ENDPT
      DO 101 J = 1, 9                                                  ENDPT
  101 AF(I, J) = 0                                                     ENDPT
C                   IS THIS THE LAST ARC                               ENDPT
      IF(IARC .NE. NARC) GO TO 104                                     ENDPT   104
C                   YES.  ADD PAYOFF PARTIALS TO BF VECTOR.            ENDPT
      IF(IPAY .NE. 9) GO TO 1011                                       SEP15   1011
      BF(8) = BF(8) + 1.                                               SEP15
      GO TO 104                                                        SEP15          104
 1011 CONTINUE                                                         SEP15
      IF(IPAY .GT. 11) GO TO 102                                       ENDPT   102
      KODE = MAP(IPAY)                                                 ENDPT
      BF(KODE) = BF(KODE) - SGN                                        ENDPT
      GO TO 104                                                        ENDPT          104
  102 ISKIP = 0                                                        ENDPT
      CALL PDBCQL(IPAY, VAL, SPART, RUMMY, 1, ISKIP)                   ENDPT
      DO 103 I = 1, 9                                                  ENDPT
  103 BF(I) = BF(I) - SGN*SPART(I)                                     ENDPT
C                   STORE THE NUMBER OF STATE TARGET CONDS. AT THIS PT. ENDPT
C                   AND SET INDEXES.                                   ENDPT
  104 MTARG = JTAB(IARC)                                               ENDPT
      IF = MTARG + 1                                                   ENDPT
      IL = 9 - MTARG                                                   ENDPT
      IP = L + MTARG                                                   ENDPT
C                   ARE THERE ANY STATE TARGETS.                       ENDPT
      IF(MTARG .GT. 0) GO TO 106                                       ENDPT   106
C                   NO.  THE TRANSVERSALITY CONDS. ARE SIMPLY THE BF   ENDPT
C                   VECTOR                                             ENDPT
      DO 105 I = 1, 9                                                  ENDPT
      K = L + I                                                        ENDPT
  105 PZI(K) = BF(I)                                                   ENDPT
      GO TO 113                                                        ENDPT          113
C                   SET UP THOSE ROWS OF THE AF MATRIX THAT RESULT FROM ENDPT
C                   THE STATE TARGETS.  AT THE SAME COMPUTE THE TARGET  ENDPT
C                   MISSES AND THEIR PARTIALS W/RESP. TO C*S.           ENDPT
  106 DO 111 I = 1, 9, MTARG                                           ENDPT
      K = L + I                                                        ENDPT
```

```
      KODE = ITC(I, IARC)                                               ENDPT
      IF(KODE .GT. 11) GO TO 108                                        ENDPT
C                    SIMPLE RELATIVE STATE MATCH                        ENDPT
      KODE = MAP(KODE)                                                  ENDPT
      AF(I, KODE) = 1                                                   ENDPT
C                    TARGET MISS                                        ENDPT
      PZI(K) = NOM(KODE) - VALTC(I, IARC)                               ENDPT
C                    PARTS. W/RESP. TO C*S OF TARGET MISS.              ENDPT
      DO 107 J = 1, NOCK                                                ENDPT
  107 DPZIDC(K, J) = S(KODE, J + 1)                                     ENDPT
      GO TO 111                                                         ENDPT
C                    COMPLEX CONDITION ON THE STATE                     ENDPT
  108 ISKIP = 0                                                         ENDPT
      CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                    ENDPT
      DO 109 J = 1, 9                                                   ENDPT
  109 AF(I, J) = SPART(J)                                               ENDPT
C                    TARGET MISS                                        ENDPT
      PZI(K) = VAL - VALTC(I, IARC)                                     ENDPT
      CALL MATMLT(TEMP, SPART, S(1,2), 1, 18, NOCK)                     ENDPT
C                    PARTS. W/RESP TO C*S OF TARGET MISS.               ENDPT
      DO 110 J = 1, NOCK                                                ENDPT
  110 DPZIDC(K, J) = TEMP(J)                                            ENDPT
  111 CONTINUE                                                          ENDPT
C                    COMPLETE AND INVERT AF MATRIX.                     ENDPT
      CALL BASIS(AF, MTARG, 9)                                          ENDPT
      CALL GJRV(AF, 9, 1.E-12, IERR)                                    ENDPT
C                    MAKE SURE INVERSION OK.                            ENDPT
      IF(IERR .NE. 0) CALL ERROR(ENDPX, -1, 1)                          ENDPT
C                    EVAL. TRANSVERSALITY CONDITIONS.                   ENDPT
      CALL MATMLT(VAL, BF, AF(1, IF), 1, 9, IL)                         ENDPT
      DO 112 I = 1, IL                                                  ENDPT
      K = IP + I                                                        ENDPT
  112 PZI(K) = VAL(I)                                                   ENDPT
C                    EVAL. PARTS. W/RESP  TO C*S OF TRANS. CONDS. BY    ENDPT
C                    DIVIDED NUMERICAL DIFFERENCES.                     ENDPT
  113 DO 124 I = 1, NOCK                                                ENDPT
C                    COMPUTE STATE/COSTATE PERT. RESULTING FROM C PERT. ENDPT
      CALL MATMLT(DZ, S(1, I + 1), DC, 18, 1, 1)                        ENDPT
C                    ADD STATE/COSTATE PERT. TO BASE VALUE.             ENDPT
      CALL MATADD(NOM, Z, DZ, 18, 1)                                    ENDPT
      CALL FORCES                                                       ENDPT
C                    RECOMPUTE BF VECTOR                                ENDPT
      DO 114 J = 1, 9                                                   ENDPT
      BF(J) = NOM(J + 9)                                                ENDPT
      DO 114 K = 1, 9                                                   ENDPT
  114 AF(J, K) = 0                                                      ENDPT
      IF(IARC .NE. NARC) GO TO 117                                      ENDPT
      IF(IPAY .NE. 9) GO TO 1141                                        SEP15
      BF(8) = BF(8) + 1.                                                SEP15
      GO TO 117                                                         SEP15
 1141 CONTINUE                                                          SEP15
      IF(IPAY .GT. 11) GO TO 115                                        ENDPT
      KODE = MAP(IPAY)                                                  ENDPT
      BF(KODE) = BF(KODE) - SGN                                         ENDPT
      GO TO 117                                                         ENDPT
  115 ISKIP = 0                                                         ENDPT
      CALL PDBCOL(IPAY, VAL, SPART, RUNNY, 1, ISKIP)                    ENDPT
      DO 116 J = 1, 9                                                   ENDPT
  116 BF(J) = BF(J) - SGN*SPART(J)                                      ENDPT
C                    ANY STATE TARGETS                                  ENDPT
  117 IF(MTARG .GT. 0) GO TO 119                                        ENDPT
C                    NO.  JUST TAKE DIVIDED DIFF. OF BF VECTOR          ENDPT
      DO 118 J = 1, 9                                                   ENDPT
      K = L + J                                                         ENDPT
  118 DPZIDC(K, I) = (BF(J) - PZI(K))/DC                                ENDPT
      GO TO 124                                                         ENDPT
C                    RECOMPUTE AF INVERSE                               ENDPT
  119 DO 122 J = 1, MTARG                                               ENDPT
      KODE = ITC(J, IARC)                                               ENDPT
      IF(KODE .GT. 11) GO TO 120                                        ENDPT
      KODE = MAP(KODE)                                                  ENDPT
      AF(J, KODE) = 1                                                   ENDPT
```

```
      GO TO 122                                                         ENDPT
  120 ISKIP = 0                                                         ENDPT
      CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                    ENDPT
      DO 121 K = 1, 9                                                   ENDPT
  121 AF(J, K) = SPART(K)                                               ENDPT
  122 CONTINUE                                                          ENDPT
      CALL BASIS(AF, MTARG, 9)                                          ENDPT
      CALL GJRV(AF, 9, 1.E-12, IERR)                                    ENDPT
C                   RECOMPUTE TRANS. CONDS.                             ENDPT
      CALL MATMLT(VAL, BF, AF(1, IF), 1, 9, IL)                         ENDPT
C                   COMPUTE DIVIDED DIFFS.                              ENDPT
      DO 123 J = 1, IL                                                  ENDPT
      K = IP + J                                                        ENDPT
  123 DPZIDC(K, I) = (VAL(J) - PZI(K))/DC                               ENDPT
  124 CONTINUE                                                          ENDPT
C                   ADD 9 TO THE NUMBER OF STATE/COSTATE TARGET MISSES  ENDPT
C                   COMPUTED.                                           ENDPT
      L = L + 9                                                         ENDPT
      RETURN                                                            ENDPT
      END                                                               ENDPT
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DC	Δc_I	I	Small perturbation of a c.	/EVAL /(	867)	BNDRY	D	DC
						BRANPT	I	DC
						ENDPT	I	DC
						INTRPT	I	DC
DZ	$\Delta c_I h_I(I^-)$	I	An 18 word array that contains the second term on the right hand side of Equation 17.4-11 of Vol.I of this document.	/EVAL /(	849)	BRANPT	I	DZ
						ENDPT	I	DZ
						INTRPT	I	DZ
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND	O	ITC
						BRANPT	I	ITC
						CHECK	I	ITC
						COSTAB	I	ITC
						COSTAI	I	ITC
						ENDPT	I	ITC
						INTRPT	I	ITC
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB
L		M	Total number of target conditions to satisfy in the problem.	/EVAL /(	868)	BNDRY	M	L
						BRANPT	M	L
						ENDPT	M	L
						INTRPT	M	L
MAP		I	A 10 word array that maps the steepest descent state vector into the QL state vector.	/EVAL /(	20)	BNDRY	D	MAP
						BRANPT	I	MAP
						ENDPT	I	MAP
						INTRPT	I	MAP
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRO	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						QLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NOCK	m_I	I	The number of c's in the vector C_I defined by Equation 17.4-4 of Vol.I of this document.	/EVAL /(	70)	BNDRY	M	NOCK
						BRANPT	I	NOCK
						ENDPT	I	NOCK
						INTRPT	I	NOCK

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NOM	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1 AL4 AL7 AL8 AL9 BCOND BNDRY BRANPT CONTRL ENDPT ENVPRQ FETCH INTERP INTRPT NLDRV NLDRV OUTPUT PDBCQL STATEF WRAPUP	I I I I I I O M I I I O M M O I I I I I	V V V V V NOM NOM NOM V NOM V NOM V NOM NOM V V V V V
PZI		M	A 40 word array that contains the target condition misses for all the target conditions in the problem.	/EVAL	/(30)	BNDRY BRANPT ENDPT INTRPT	I M M M	PZI PZI PZI PZI
S		I	An 18x41 array used to store the particular and homogeneous solutions on the early side of a corner point.	/EVAL	/(71)	BNDRY BRANPT ENDPT INTRPT	I I I I	S S S S
SGN		I	Sign of the variable SIG in the 65th word of common block /GOBAL/. SGN = +: payoff to be maximized; SGN = -: payoff to be minimized.	/EVAL	/(1)	BNDRY ENDPT	O I	SGN SGN
SPART		I	An 18 word array whose first nine entries receive the values of the partial derivatives wrt the state of those target conditions computed in subroutine PDBCQL.	/EVAL	/(2)	BNDRY BRANPT ENDPT INTRPT	O I I I	SPART SPART SPART SPART
TEMP	$(\partial\psi_I/\partial C_I)^T$	I	A 40 word array that contains the transpose of the vector defined by Equation 17.4-9 of Vol.I of this document.	/EVAL	/(809)	BRANPT ENDPT INTRPT	I I I	TEMP TEMP TEMP
VALTC		I	A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK	/(1062)	BCOND BRANPT CHECK ENDPT INTRPT	O I I I I	VALTC VALTC VALTC VALTC VALTC
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY BRANPT ENDPT ENVPRQ INTERP INTRPT LINDRV NORMAL OUTPUT RKUTT1 RKUTT2 SALVE WRAPUP	I I I I O I I M I O M M M	Z Z Z Z Z Z Z Z Z Z Z Z Z

SUBRØUTINE
ENVPRQ

1077

Purpose

ENVPRQ calculates and stores environmental trajectory parameters from a converged QL trajectory for use in the SSSP sizing module.

ENVPRQ

```
      SUBROUTINE ENVPRQ                                                 JUL21
C           COMPUTE   ENVIRONMENTAL  PARAMETERS THAT AFFECT  WEIGHT     JUL21
      DATA RAD/57.2957795130823/                                        JUL21
      COMMON/GLOBAL/                                                    GLOBAL
     *GR     ,ER     ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER , GLOBAL
     *ITRMAX,JJOP(6)       ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /ZD/ ZD(50)                                                ZD
      COMMON /Z/ Z(50)                                                  Z
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C            PHASE II SIZING PARAMERERS                                 SIZING
     *TZ,      VV(3),      QP(14),    EROR,     PZ(5),      VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,     TLNG,                   SIZING
C          PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,      WLOO,      DWEB,      DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,     NNB,       WEO,       WEB,       WO,       WLO,    SIZING
     *DVO,      DVB,       MUB,       MUO,       VSTG,      WPO               SIZING
     * JTYP,    BECO,      BSTG,      ORBI,      ITNBW ,  ITNOW   ,ISZO(23)  SIZING
      EQUIVALENCE (SE(6),QLIM)                                          JUL21
      IF(INTB.EQ.2)  GO TO  100                                         JUL21  100
C       I         MAXIMUM  DYNAMIC PRESSURE                             JUL21
   10 IF( IARC.GE.IFIX(SQ(1,3))) RETURN                                 JUL21
      IF( Q.LE. QLIM    )  RETURN                                       JUL21
      QLIM = Q                                                          JUL21
      SQ(5,1) = TIME                                                    JUL21
      SQ(6,1) = R-ER                                                    JUL21
      SQ(7,1)= V                                                        JUL21
```

```
          SQ(8,1)= GAM*RAD                                                        JUL21
          SQ(9,1) = MACH                                                          JUL21
          RETURN                                                                  JUL21
C         II  BRANCH  TRAJECTORY TEST                                             JUL21
    100   IF(IARC.LE.NSB) GO TO 10                                                JUL21
          IF(IFIX(SQ(1,3)).LE NSB+NSAB)  GO TO 120                                JUL21
          IF(IARC.GT.NSB.AND IARC LE.NSB+NSAB) GO TO 200                          JUL21
          RETURN                                                                  JUL21
    120   IF(IARC.GT.NSB+NSAB) GO TO 200                                          JUL21
          RETURN                                                                  JUL21
C           III     ENTRY  TRAJECTORY  PARAMETERS  Q*SLPHA AND HEAT RATE          JUL21
    200   CONTINUE                                                                JUL21
          QALF = Q*ALPHA*RAD                                                      JUL21
          IF(QALF.GT.SV(1)   )    SV(1)  = QALF                                   JUL21
          TLOFAC =  SQRT(LIFT*LIFT +DRAG*DRAG) / W                                JUL21
          IF(TLOFAC.GT.SV(27)) SV(27)= TLOFAC                                     JUL21
C      IIIA THRESHOLD    HEAT  LOAD    AND  TIME                                  JUL21
          IF(THRESH.GT.0) GO TO  220                                              JUL21
          IF(ZD(9).LT. SQ(30,2)) RETURN                                           JUL21
          THRESH = TIME                                                           JUL21
          HT1=Z(9)                                                                JUL21
          RETURN                                                                  JUL21
    220   IF(ZD(9).LT.SQ(30,2)) RETURN                                            JUL21
          SQ(30,3)= TIME -THRESH                                                  JUL21
          SQ(30,4)= Z(9)-HT1                                                      JUL21
          RETURN                                                                  JUL21
          ENTRY QLNVPI                                                            JUL21
          IF(NFARC EQ.NARC) GO TO 7                                               JUL21
          INTB=2                                                                  JUL21
          NSB=NBRAN                                                               JUL21
          NSAB=NFARC-NSB                                                          JUL21
          GO TO 8                                                                 JUL21
      7   INTB=0                                                                  JUL21
      8   CONTINUE                                                                JUL21
          THRESH =0.                                                              JUL21
          RETURN                                                                  JUL21
          END                                                                     JUL21
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	I	Angle of attack	(RAD)	/DYNA /(	79)	AEROCO	I	ALPHA
							ALGCON	M	ALPHA
							AL2	I	ALPHA
							ARCIN	M	ALPHA
							CONTRL	M	ALPHA
							ENVPRQ	I	ALPHA
							MOMECO	I	ALPHA
							NPLANE	M	ALPHA
							OUTPUT	I	ALPHA
							TRAJIN	O	ALPHA
							UT	I	ALPHA
							WRAPUP	I	ALPHA
DRAG	D	I	Aerodynamic drag	(LBS)	/DYNA /(	69)	AL5	I	DRAG
							AL7	I	DRAG
							AL8	I	DRAG
							AL9	I	DRAG
							APPLY	I	DRAG
							CONTRL	I	DRAG
							ENVPRQ	I	DRAG
							NLDRV	I	DRAG
							OUTPUT	I	DRAG
							TH3	I	DRAG
							UT	M	DRAG
ER	E_R	I	Earth radius.	(FT)	/GLOBAL/(	2)	ENVPRQ	I	ER
							PDBCOL	I	ER
							QLTOSZ	I	ER
							STATEF	I	ER
GAM	γ	I	Relative flight path angle.	(RAD)	/D /(	92)	ARCIN	I	GAM
							ENVPRQ	I	GAM
							OUTPUT	I	GAM
							STATEF	I	GAM
							WRAPUP	I	GAM
LIFT	L	I	Aerodynamic lift	(LBS)	/DYNA /(	60)	AL4	I	LIFT
							AL5	I	LIFT
							AL6	I	LIFT
							APPLY	I	LIFT
							CONTRL	I	LIFT
							ENVPRQ	I	LIFT
							OUTPUT	I	LIFT
							TH3	I	LIFT
							UT	O	LIFT
MACH	M	I	Mach number		/DYNA /(	26)	AEROCO	I	MACH
							ENVPRQ	I	MACH
							OUTPUT	I	MACH
							STATEF	M	MACH
NARC	N_3	I	Number of subarcs in the problem.		/GLOBAL/(	18)	BCOND	I	NARC
							BNDRY	I	NARC
							CHECK	I	NARC
							ENDPT	I	NARC
							ENVPRQ	I	NARC
							FETCH	I	NARC
							INARC	I	NARC
							MAGIC	I	NARC
							QLTOSZ	I	NARC
							SALVE	I	NARC
							WRAPUP	I	NARC
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.		/GLOBAL/(	19)	BNDRY	I	NBRAN
							BRANPT	I	NBRAN
							COSTAB	I	NBRAN
							ENVPRQ	I	NBRAN
							INTRPT	I	NBRAN
							MAGIC	I	NBRAN
							QLTOSZ	I	NBRAN
							SALVE	I	NBRAN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND BNDRY BRANPT COSTAB ENVPRO INTRPT MAGIC OLTOSZ SALVE	I I I I I I I I I	NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC NFARC
Q	q	I	Dynamic pressure (LBS/FT^2)	/DYNA /(	27)	ENVPRO OUTPUT PDBCOL STATEF UT	I I I M I	Q Q Q Q Q
QLIM		M	Saved value of maximum dynamic pressure.	/SIZING/(	264)	ENVPRO	M	QLIM
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA /(	7)	AL4 AL7 AL8 AL9 CONTRL ENVPRO NLDRV PDBCOL OLTOSZ STATEF	I I I I I I I I I M	R R R R R R R R R R
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRO OLTOSZ	M M	SQ SQ
SV		M	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRO OLTOSZ	M M	SV SV
TIME		I	Trajectory time (SEC)	/DYNA /(	2)	ENVPRO OUTPUT PDBCOL STATEF WRAPUP	I I I M I	TIME TIME TIME TIME TIME
V	V	I	Relative velocity. (FT/SEC)	/D /(	91)	AL1 AL4 AL7 AL8 AL9 BCOND BNDRY BRANPT CONTRL ENDPT ENVPRO FETCH INTERP INTRPT NLDRV NLDRV OUTPUT PCBCOL STATEF WRAPUP	I I I I I I O M I I I O M M O I I I I I	V V V V V NOM NOM NOM V NOM V NOM V NOM NOM V V V V V
W	W	I	Weight (LBS)	/DYNA /(	91)	AL5 ENVPRO OUTPUT PDBCOL OLTOSZ STATEF TH3	I I I I I M I	W W W W W W W

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRV	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRQ	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZD		I	A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.I of this document.	/ZD	/(1)	ENVPRQ	I	ZD
						LINDRV	I	ZD
						OUTPUT	I	ZD
						RKUTT2	I	ZD
						WRAPUP	I	ZD

SUBRØUTINE
ERRØR

1084

Purpose

ERROR is a general error message routine for the QL module.

```
      SUBROUTINE ERROR(REGION, KOD, IRETRN)                             ERROR
C                                                                       ERROR
C                     THIS ROUTINE SERVES AS A GENERAL ERROR MESSAGE ROU-  ERROR
C     TINE FOR THE QTINE FOR THE QL MODULE   REGION IS THE 6 HOLLERITH  ERROR
C                     CHARACTER NAME OF THE SUBROUTINE IN WHICH THE ERROR  ERROR
C                     OCURRED, KOD IS THE CODE NO. OF THE ERROR AND IRETRN ERROR
C                     INDICATES WHETHER THE PROBLEM SHOULD BE RESTARTED  ERROR
C                     WITH A BETTER STEEP. DESC. INITIAL ARC.           ERROR
C                                                                       ERROR
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *ATT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
    1 FORMAT(1H0, 5H*****, A6, 10H ERROR NO., I2, 5H*****)             ERROR
C                     PRINT ERROR MESSAGE                               ERROR
      K = IABS(KOD)                                                     ERROR
      WRITE(6, 1) REGION, K                                             ERROR
C                     IF KOD POSITIVE,WARNING ONLY.                     ERROR
      IF(KOD) 101,101,102                                               ERROR   101  102
  101 KGLOBL = 1                                                        ERROR
      CALL PDUMP(X, DELT(20),1,XX, IDAM, 1)                             ERROR
C                     SHOULD STEEP. DESC. TRY RESTART.                  ERROR
C                     FATAL QL ERROR.                                   ERROR
      IF(IRETRN .NE. 0) IFATAL = .TRUE.                                 ERROR
      CALL QLERR                                                        ERROR
  102 RETURN                                                            ERROR
      END                                                               ERROR
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DELT		I	A twenty word array containing the quasitime compute interval for each subarc.	/D	/(211)	CHECK	O	DELT
						ERROR	I	DELT
						INARC	I	DELT
IDAM		I	Optional atmospheric calculations flag. IDAM = -1: Compute $\partial^3\rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3\rho_a/\partial R^3$, μ_a, $\partial\mu_a/\partial R$, etc.	/DYNA	/(218)	ARCIN	O	IDAM
						ERROR	I	IDAM
						NPLANE	O	IDAM
						STATEF	I	IDAM
						WRAPUP	O	IDAM
IFATAL		O	Fatal error flag.	/GLOBAL/	(17)	ERROR	O	IFATAL
KGLOBL		O	A seven word array not used.	/GLOBAL/	(96)	ERROR	O	KGLOBL
X	x	I	The quasitime variable.	/D	/(1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
XX		I	Fraction of subarc that has transpired	/DYNA	/(1)	ARCIN	O	XX
						ERROR	I	XX
						OUTPUT	I	XX
						STATEF	M	XX

SUBRØUTINE
FETCH

Purpose

FETCH retrieves the state/costate vector from the initial arc at the early and/or late sides of corner points.

FETCH

```
      SUBROUTINE FETCH(IARC)                                            FETCH
C                                                                       FETCH
C                    THIS ROUTINE RETRIEVES THE STATE/COSTATE VECTOR    FETCH
C                    FROM THE INITIAL ARC AT THE EARLY AND/OR LATE SIDES FETCH
C                    OF CORNER PTS.                                     FETCH
C                                                                       FETCH
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YAURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      DIMENSION SCS(16), QTP(20), MAP(16), SKIP(3)                      FETCH
      DATA FETCHX/6HFETCH /, IPASS/0/, IARCS/20/                        FETCH
      DATA MAP                                                          FETCH
     */1, 2, 4, 7, 3, 5, 6, 9, 10, 11, 13, 16, 12, 14, 15, 18/          FETCH
C                    IS THIS PT. ALREADY IN CORE.                       FETCH
      IF(IARC .EQ. IARCS) RETURN                                        FETCH
C                    NO.  IS THIS THE FIRST TIME FETCH HAS BEEN CALLED. FETCH
      IF(IPASS .NE. 0) GO TO 101                                        FETCH
C                    YES.  INITIALIZE INITIAL ARC FILE.                 FETCH
      REWIND INARK                                                      FETCH
      READ(INARK) TO, NS, NARCTP, (QTP(IJ), IJ = 1, NARCTP)             FETCH
      IF(NARCTP .NE. NARC) CALL ERROR(FETCHX, -1, 1)                    FETCH
      READ(INARK) KARC, TIME, SKIP, SCS                                 FETCH
      IPASS = 1                                                         FETCH
C                    INITIALIZE LOGICAL PARAMETERS                      FETCH
  101 ISGN = 1                                                          FETCH
      JARC = 0                                                          FETCH
      IARCS = IARC                                                      FETCH
C                    IS THIS A CALL FOR THE INITIAL PT. OF THE FILE.    FETCH
      IF(IARC .EQ. 0) GO TO 100                                         FETCH
C                    NO.  SET UP LOGICAL PARAMETERS.                    FETCH
      JARC = IABS(IARC)                                                 FETCH
      ISGN = ISIGN(1, IARC)                                             FETCH
C                    POSITION FILE TO DESIRED PT.                       FETCH
  100 IF(JARC .GT. KARC) GO TO 102                                      FETCH
      REWIND INARK                                                      FETCH
      READ(INARK) TO                                                    FETCH
  102 READ(INARK) KARC, TIME, SKIP, SCS                                 FETCH
      IF(EOF, INARK) 103, 109                                           FETCH
  103 IF(JARC .EQ. NARC .AND. ISGN .LT. 0) GO TO 104                    FETCH
      CALL ERROR(FETCHX, -2, 1)                                         FETCH
  104 BACKSPACE INARK                                                   FETCH
      BACKSPACE INARK                                                   FETCH
      READ(INARK) KARC, TIME, SKIP, SCS                                 FETCH
C                    SET UP STATE/COSTATE VECTOR AND COMPUTE ANY OTHER  FETCH
C                    QUANTITES THAT MAY BE NEEDED.                      FETCH
  105 DO 106 I = 1, 16                                                  FETCH
      J = MAP(I)                                                        FETCH
  106 NOM(J) = SCS(I)                                                   FETCH
      IF(KARC - 1) 107, 107, 108                                        FETCH
  107 NOM(8) = QTP(KARC) - TO                                           FETCH
      GO TO 110                                                         FETCH
```

```
      108 NOM(8) = QTP(KARC) - QTP(KARC - 1)                                  FETCH
          GO TO 110                                                           FETCH
      109 IF(KARC .LE. JARC) GO TO 102                                        FETCH
          IF(ISGN .GT. 0) GO TO 105                                           FETCH
          GO TO 104                                                           FETCH
      110 X = JARC                                                            FETCH
          LARC = JARC                                                         FETCH
          IF(ISGN .GT. 0) LARC = JARC + 1                                     FETCH
          CALL READMS(9, ARCDA, 42, LARC)                                     FETCH
          CALL FORCES                                                         FETCH
          RETURN                                                              FETCH
          END                                                                 FETCH
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ARCDA	S_{ref}	I	Aerodynamic reference area	(FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
							BNDRY	I	ARCDA
							CHECK	I	ARCDA
							FETCH	I	ARCDA
							SALVE	I	ARCDA
							STATEF	I	SREF
							UT	I	SREF
							WRAPUP	I	ARCDA
INARK		I	Logical unit on which initial and converged arcs are stored. INARK = 11.		/GLOBAL/(	95)	CHECK	O	INARK
							FETCH	I	INARK
							INARC	I	INARK
							MARCH	I	INARK
							WRAPUP	I	INARK
NARC	N_3	I	Number of subarcs in the problem.		/GLOBAL/(	18)	BCOND	I	NARC
							BNDRY	I	NARC
							CHECK	I	NARC
							ENDPT	I	NARC
							ENVPRQ	I	NARC
							FETCH	I	NARC
							INARC	I	NARC
							MAGIC	I	NARC
							OLTOSZ	I	NARC
							SALVE	I	NARC
							WRAPUP	I	NARC
NOM	V	O	Relative velocity.	(FT/SEC)	/D /(	91)	AL1	I	V
							AL4	I	V
							AL7	I	V
							AL8	I	V
							AL9	I	V
							BCOND	I	NOM
							BNDRY	O	NOM
							BRANPT	M	NOM
							CONTRL	I	V
							ENDPT	I	NOM
							ENVPRQ	I	V
							FETCH	O	NOM
							INTERP	M	V
							INTRPT	M	NOM
							NLDRV	O	NOM
							NLDRV	I	V
							OUTPUT	I	V
							PDBCOL	I	V
							STATEF	I	V
							WRAPUP	I	V
TO	t_0	M	Trajectory start time.	(SEC)	/GLOBAL/(	7)	FETCH	M	TO
							INARC	M	TO
							TRAJIN	I	TO
							WRAPUP	I	TO
X	x	O	The quasitime variable.		/D /(	1)	AL4	I	X
							BNDRY	O	X
							ERROR	I	X
							FETCH	O	X
							FORCES	I	X
							INARC	M	X
							INTERP	I	X
							MADAMS	M	X
							RKUTT1	M	X
							RKUTT2	M	X
							SALVE	M	X
							STATEF	I	X
							WRAPUP	M	TT

SUBRØUTINE
FØRCES

Purpose

FORCES controls the calculation of all quantities needed by subroutine APPLY.

```
      SUBROUTINE FORCES                                               FORCES
C                                                                     FORCES
C                   THIS ROUTINE CONTROLS THE COMPUTATION OF ALL QUAN- FORCES
C                   TITIES NEEDED TO COMPUTE THE APPLIED ACCELERATIONS FORCES
C                   ON THE VEHICLE.                                   FORCES
C                                                                     FORCES
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM D
     *, LHT                                                           D
      COMMON /D/                                                      D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20) D
      DIMENSION NOM(20)                                               D
      EQUIVALENCE (NOM, V)                                            D
      COMMON /CNTRL/                                                  CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   , CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES , CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT, CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES        CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                  CNTRL
      LOGICAL SWITCH, ILOAD                                           DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,     DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,   DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                             DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                               DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                      DYNA
      COMMON /DYNA/                                                   DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  , JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   , DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     , DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM , DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   , DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR , DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA , DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA , DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM , DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   , DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR, DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  , DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR, DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  , DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR, DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA, DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   , DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX , DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED  DYNA
      COMMON /DYNA/                                                   DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF , DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  , DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198, DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA, DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216, DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   , JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                               AUG09
C                   IS THIS THE FIRST PT. OF THE ARC.                 FORCES
      IF(KPT .NE. 1) GO TO 101                                        FORCES
      IF(ABS(X - FLOAT(IFIX(X))) .NE. 0.) GO TO 101                   FORCES
C                   YES.  IS THIS THE FIRST SUBARC.                   FORCES
      IF(IARC .EQ. 1) CALL TRAJIN                                     FORCES
      CALL ARCIN                                                      FORCES
  101 CALL STATEF                                                     FORCES
C                   ARE THERE ANY APPLIED LOADS AT ALL                FORCES
      IF(KODE .EQ. 0) GO TO 102                                       FORCES
      CALL CONTRL                                                     FORCES
C                   IS THIS POWERED FLIGHT                            FORCES
      IF(IPOW .GT. 0 .AND. J1 .NE. 4) CALL IMPULS                     JUL21
C                   IS THIS THE LAST PT. OF SUBARC                    FORCES
  102 IF(KPT .EQ. NPTS) CALL ARCEN                                    FORCES
      RETURN                                                          FORCES
      END                                                             FORCES
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IPOW		I	Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag	/DYNA /(	139)	ARCIN	M	IPOW
						FORCES	I	IPOW
						NPLANE	I	IPOW
						STATEF	I	IPOW
						THROTL	I	IPOW
J1		I	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY	I	J1
						ARCIN	O	J1
						CONTRL	M	J1
						FORCES	I	J1
						NPLANE	I	J1
						STATEF	I	J1
						THROTL	M	J1
KODE		I	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.	/DYNA /(	25)	APPLY	I	KODE
						ARCIN	O	KODE
						CONTRL	M	KODE
						FORCES	I	KODE
						NLDRV	I	KODE
						STATEF	I	KODE
KPT		I	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND	O	KPT
						BNDRY	O	KPT
						FORCES	I	KPT
						MAGIC	O	KPT
						RKUTT1	I	KPT
						SALVE	M	KPT
						WRAPUP	M	KPT
NPTS		I	The total number of points in the subarc.	/CNTRL /(	19)	BCOND	O	NPTS
						BNDRY	O	NPTS
						FORCES	I	NPTS
						INARC	M	NPTS
						MAGIC	O	NPTS
						SALVE	M	NPTS
						WRAPUP	O	NPTS
X	x	I	The quasitime variable.	/D /(	1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT

Purpose

GJRV is a general matrix inversion routine.

```
      SUBROUTINE GJRV(A,N,EPSIL,IERR)                                        GJRV
CGJRV    GAUSS-JORDAN INVERSION OF MATRICES PROPOSED BY H.RUTISHAUSER        GJRV
C        IN ALGOL. ACM JOURNALL FEB. 1962. FORTRAN BY RAGUSA.                GJRV
      DIMENSION A(N, N), B(40), C(40), IP(40), IQ(40)                        GJRV
      IERR=0                                                                 GJRV
 1001 DO 140K=1,N                                                            GJRV
      PIVOT=0.0                                                              GJRV
      DO 120 I=K,N                                                           GJRV
      DO 2 J=K,N                                                             GJRV
        IF(ABS (A(I,J))-ABS (PIVOT)) 2,2,1                                   GJRV
    1 PIVOT=A(I,J)                                                           GJRV
      IP(K)=I                                                                GJRV
      IQ(K)=J                                                                GJRV
    2 CONTINUE                                                               GJRV
  120   CONTINUE                                                             GJRV
      IF(ABS (PIVOT)-EPSIL)100,100,3                                         GJRV
    3 IF(IP(K)-K)4,6,4                                                       GJRV
    4 DO 5 J=1,N                                                             GJRV
      IPX=IP(K)                                                              GJRV
      Z=A(IPX,J)                                                             GJRV
      A(IPX,J)=A(K,J)                                                        GJRV
    5 A(K,J)=Z                                                               GJRV
    6 IF(IQ(K)-K)7,9,7                                                       GJRV
    7 DO 8 I=1,N                                                             GJRV
      IPX=IQ(K)                                                              GJRV
      Z=A(I,IPX)                                                             GJRV
      A(I,IPX)=A(I,K)                                                        GJRV
    8 A(I,K)=Z                                                               GJRV
    9 DO 13 J=1,N                                                            GJRV
      IF(J-K)11,10,11                                                        GJRV
   10 B(J)=1.0/PIVOT                                                         GJRV
      C(J)=1.0                                                               GJRV
      GO TO 12                                                               GJRV
   11 B(J)=-A(K,J)/PIVOT                                                     GJRV
      C(J)=A(J,K)                                                            GJRV
   12 A(K,J)=0.0                                                             GJRV
      A(J,K)=0.0                                                             GJRV
   13 CONTINUE                                                               GJRV
      DO 14 I=1,N                                                            GJRV
      DO 14 J=1,N                                                            GJRV
   14 A(I,J)=A(I,J)+C(I)*B(J)                                                GJRV
  140     CONTINUE                                                           GJRV
      DO 20 KP=1,N                                                           GJRV
      K=N+1-KP                                                               GJRV
       IF(IP(K)-K) 15,17,15                                                  GJRV
   15 DO 16 I=1,N                                                            GJRV
      IPX=IP(K)                                                              GJRV
      Z=A(I,IPX)                                                             GJRV
      A(I,IPX)=A(I,K)                                                        GJRV
   16 A(I,K)=Z                                                               GJRV
   17 IF(IQ(K)-K)18,20,18                                                    GJRV
   18 DO 19 J=1,N                                                            GJRV
      IPX=IQ(K)                                                              GJRV
      Z=A(IPX,J)                                                             GJRV
      A(IPX,J)=A(K,J)                                                        GJRV
   19 A(K,J)=Z                                                               GJRV
   20 CONTINUE                                                               GJRV
      GO TO 21                                                               GJRV
  100 IERR=-1                                                                GJRV
   21 RETURN                                                                 GJRV
       END                                                                   GJRV
```

1Ø48

SUBRØUTINE IMPULS

1004

Purpose

IMPULS computes the rocket's specific impulse as a function of net thrust. In addition, it computes

$$\frac{\partial I_{sp}}{\partial T} \quad \text{and} \quad \frac{\partial^2 I_{sp}}{\partial T^2}$$

```
       SUBROUTINE IMPULS                                                IMPULS
C                                                                       IMPULS
C                    THIS ROUTINE COMPUTES THE VACUUM SPECIFIC IMPULSE  IMPULS
C                                                                       IMPULS
       COMMON/ARCDAT/                                                   ARCDAT
      *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI     , ARCDAT
      *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX     , ARCDAT
      *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA     , ARCDAT
      *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG     , ARCDAT
      *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB     , ARCDAT
      *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT       , ARCDAT
      *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE      ARCDAT
       DIMENSION  ARCDA(40)                                             ARCDAT
       EQUIVALENCE(SREF,ARCDA)                                          ARCDAT
       LOGICAL SWITCH, ILOAD                                            DYNA
       REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,     DYNA
      *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,    DYNA
      *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
      *IRATED, ISPF, ISPFF                                              DYNA
       REAL MACHV, MACHR, MACHVR, MACHRR                                DYNA
       REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                       DYNA
       COMMON /DYNA/                                                    DYNA
      *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,  JUL21
      *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,  DYNA
      *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,  DYNA
      *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,  DYNA
      *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,  DYNA
      *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,  DYNA
      *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,  DYNA
      *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,  DYNA
      *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,  DYNA
      *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,  DYNA
      *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,  DYNA
      *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,  DYNA
      *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,  DYNA
      *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,  DYNA
      *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,  DYNA
      *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,  DYNA
      *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,  DYNA
      *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,  DYNA
      *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED   DYNA
       COMMON /DYNA/                                                    DYNA
      *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,  DYNA
      *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,  DYNA
      *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,  DYNA
      *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,  DYNA
      *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,  DYNA
      *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,  JUL21
      *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                AUG09
C                    COMPUTE THE VACUUM THRUST                          IMPULS
      FVAC = T + AE*PA                                                  IMPULS
C                    INITIALIZE ISP AND PARTIALS W/RESP. TO THRUST      IMPULS
      ISP = XISP                                                        IMPULS
      ISPF = 0.                                                         IMPULS
      ISPFF = 0.                                                        IMPULS
C                    CHECK FOR DUAL ENGINE SIMULATION                   JUL21
      IF(JPRO .EQ. 2) GO TO 100                                         SEP15   100
C                    WERE ISP LOSSES INPUT                              IMPULS
      IF(MISP .LE. 0) RETURN                                            IMPULS
C                    YES.  COMPUTE ISP AND PARTS. W/RESP. TO THRUST     IMPULS
      FOR = FVAC/FRATED                                                 IMPULS
      CALL SPLINE(MISP, FOR, CISP, CISPF, CISPFF)                       IMPULS
   50 ISP = CISP*IRATED                                                 JUL21
      ISPF = CISPF*IRATED/FRATED                                        IMPULS
      ISPFF = CISPFF*IRATED/FRATED**2                                   IMPULS
      RETURN                                                            SEP15
C                    DUAL ENGINE SIMULATION                             JUL21
  100 FOR = FVAC/FRATED                                                 JUL21
      CALL ISPRAT(FOR, CISP, CISPFF, CISPFF, 2)                         JUL21
      GO TO 50                                                          JUL21   50
      END                                                               IMPULS
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AE	A_{exit}	I	Total nozzle exit area	(FT^2)	/DYNA /(	89)	APPLY	I	AE
							ARCIN	O	AE
							IMPULS	I	AE
							NLDRV	I	AE
							TH2	I	AE
FRATED		I	1% of the maximum rocket vacuum thrust	(LBS)	/DYNA /(	171)	ARCIN	O	FRATED
							IMPULS	I	FRATED
FVAC		M	Total vacuum thrust (rocket)	(LBS)	/DYNA /(	33)	APPLY	I	FVAC
							ARCIN	M	FVAC
							IMPULS	M	FVAC
							NLDRV	I	FVAC
							STATEF	M	FVAC
							TH2	I	FVAC
IRATED		I	1% of the maximum rated I_{SP}	(SECS)	/DYNA /(	170)	ARCIN	O	IRATED
							IMPULS	I	IRATED
ISP	I_{SP}	O	Vacuum specific impulse	(SECS)	/DYNA /(	45)	APPLY	I	ISP
							ARCIN	O	ISP
							IMPULS	O	ISP
ISPF	$\partial I_{SP}/\partial T$	O	See symbol		/DYNA /(	179)	APPLY	I	ISPF
							IMPULS	O	ISPF
ISPFF	$\partial^2 I_{SP}/\partial T^2$	O	See symbol		/DYNA /(	180)	APPLY	I	ISPFF
							IMPULS	O	ISPFF
JPRO		I	Propulsion model option flag		/ARCDAT/(	10)	ARCIN	I	JPRO
							IMPULS	I	JPRO
MISP		I	Curve number XISP loss table		/ARCDAT/(	26)	ARCIN	I	MISP
							IMPULS	I	MISP
PA	P_a	I	Atmospheric pressure	(LBS/FT^2)	/DYNA /(	14)	IMPULS	I	PA
							NLDRV	I	PA
							OUTPUT	I	PA
							TH2	I	PA
T	T	I	Thrust	(LBS)	/DYNA /(	42)	ALGCON	M	T
							AL1	I	T
							AL4	I	T
							AL6	I	T
							AL7	I	T
							AL8	I	T
							AL9	I	T
							APPLY	I	T
							ARCIN	O	T
							CONTRL	M	T
							DL2	I	T
							IMPULS	I	T
							OUTPUT	I	T
							TH1	I	T
							TH2	I	T
							TH3	I	T
							TH4	I	T
XISP	I_{sp}	I	Vacuum specific impulse	(SEC)	/ARCDAT/(	3)	ARCIN	I	XISP
							IMPULS	I	XISP

SUBRØUTINE
INARC

Purpose

INARC stores the initial arc on logical unit 12. The values stored are linearly interpolated from the initial arc file on logical unit 11.

```
      SUBROUTINE INARC                                                  INARC
C                                                                       INARC
C                   THIS ROUTINE STORES THE INITIAL ARC OUT ON UNIT 12. INARC
C                                                                       INARC
      EXTERNAL EXIT                                                     INARC
      EXTERNAL NUCASE                                                   INARC
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /MAP/ MAP(10)                                              MAP
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /Y/ Y(820)                                                 Y
      DIMENSION QTAPE(20), IMAP(20), T(100)                             INARC
      EQUIVALENCE  (T, IT)                                              INARC
      DATA XINARC /6H INARC/, RAD/.0174532925199433/                    INARC
    1 FORMAT(1H1, 25X, 24H****INITIAL ARC DATA****)                     INARC
    2 FORMAT(1H0, 25X, 23H****SCALING FACTORS****//(1H , 5(E15.7)))     INARC
    3 FORMAT(1H0, 27X, 13H****TABLE NO., I2, 4H****/)                   INARC
    4 FORMAT(1H , 5E15.7)                                               INARC
    5 FORMAT(1H0, 18X, 37H****THE INITIAL ARC IS ON UNIT 11****)        INARC
C                                                                       INARC
      WRITE(6, 1)                                                       INARC
      IERR = 0                                                          INARC
      REWIND INARK                                                      INARC
      REWIND 12                                                         INARC
      DO 105 I = 1, NUP                                                 INARC
  105 Y(I) = 0.                                                         INARC
      NARCTP = 1                                                        INARC
C                  READ IN TRAJ. START TIME, NUMBER OF ARCS AND ARC     INARC
C                  STAGING TIMES FROM 1ST RECORD.                       INARC
      READ(INARK) TOTAPE, NS, NARCTP, (QTAPE(IK), IK = 1, NARCTP)       INARC
      IF(EOF, INARK) 101, 9                                             INARC
    9 WRITE(6, 5)                                                       INARC
C                  MAKE SURE NUMBER OF ARCS AGREES WITH USER*S INPUT.   INARC
      IF(NARCTP .NE. NARC) CALL ERROR(XINARC, -4, 1)                    INARC
C                  INITIALIZE TIMES                                     INARC
      TO = TOTAPE                                                       INARC
      TSTART = TO                                                       INARC
C                  COMPUTE RECORD SIZE                                  INARC
      NWRDS = 5 + 2*NS                                                  INARC
      KERR = 0                                                          INARC
C                  SET UP TRANSLATION MAP FROM STEEPEST DESC. STATE/    INARC
C                  COSTATE ORDERING TO QL ORDERING.                     INARC
      ILIM = NS + 1                                                     INARC
      DO 11 I = 1, ILIM                                                 INARC
      K = I + MAP(I)                                                    INARC
      J = I + ILIM                                                      INARC
      IMAP(I) = K                                                       INARC
   11 IMAP(J) = K + NS                                                  INARC
      MOM = 0.                                                          INARC
      DO 40  IARC = 1, NARC                                             INARC
C                  STORE NUMBER OF HOMOGENEOUS SOLUTIONS TO BE INTE-    INARC
C                  GRATED OVER THIS ARC.                                INARC
      MOM =       NOC(IARC)                                             INARC
C                  COMPUTE RECORD SIZE ON UNIT 12 OVER THIS ARC.        INARC
```

9 101

```
      NN = N*(MOM + 1)                                                  INARC
C                    COMPUTE AND STORE TAU FOR THIS ARC                 INARC
      TAU = QTAPE(IARC)-TSTART                                          INARC
      QT(IARC) = TAU                                                    INARC
      XARC = IARC                                                       INARC
      X = IARC - 1                                                      INARC
C                    COMPUTE AND STORE NUMBER OF PTS. AND STEPSIZE FOR  INARC
C                    THIS ARC.                                          INARC
      DT = DELT(IARC)                                                   INARC
      NPTS = TAU/DT + 1.5                                               INARC
      EPS = ABS(TAU - (NPTS - 1)*DT)                                    INARC
      IF(EPS .GT. .5) NPTS = NPTS + 1                                   INARC
      IF(NPTS .LT. 4) NPTS = 4                                          INARC
      NPOINT(IARC) = NPTS                                               INARC
      H = 1./FLOAT(NPTS - 1)                                            INARC
C                    GET FIRST TWO RECORDS FOR THIS ARC INTO CORE.      INARC
      IFT = 2                                                           INARC
      LFT = 1                                                           INARC
  301 IF(KERR .EQ. 0) READ(INARK) (T(J), J = 1, NWRDS)                  INARC
      IF(IT - IARC) 301, 303, 302                                       INARC
  302 BACKSPACE INARK                                                   INARC
      BACKSPACE INARK                                                   INARC
      GO TO 301                                                         INARC
  303 MIN = NWRDS + 1                                                   INARC
      MAX = NWRDS + NWRDS                                               INARC
      IF(KERR .EQ. 0) READ(INARK) (T(J), J = MIN, MAX)                  INARC
      TLAST = T(2)                                                      INARC
      TNEXT = T(NWRDS + 2)                                              INARC
C                    PUT THIS ARC OUT ONTO UNIT 12                      INARC
      DO 30  K = 1, NPTS                                                INARC
      XX = X + 1. - XARC                                                INARC
      TIME = TSTART + XX*TAU                                            INARC
      IF(K .EQ. NPTS) TIME = TSTART + TAU                               INARC
      IF(TIME .LE. TNEXT) GO TO 12                                      INARC
      IF(KERR .EQ. 0) CALL MARCH(TIME, IFT, T, NWRDS, KERR)             INARC
      LFT = 3 - IFT                                                     INARC
      KK = 2 + NWRDS*(IFT - 1)                                          INARC
      LL = 2 + NWRDS*(LFT - 1)                                          INARC
      TLAST = T(LL)                                                     INARC
      TNEXT = T(KK)                                                     INARC
   12 DT = TNEXT - TLAST                                                INARC
      FCTR = (TIME - TLAST)/DT                                          INARC
      DO 20 I = 1, N                                                    INARC
   13 IF(I .NE. IOP) GO TO 10                                           INARC
      Y(I) = TAU                                                        INARC
      GO TO 20                                                          INARC
   10 J = IMAP(I)                                                       INARC
      KK = J + NWRDS*(IFT - 1)                                          INARC
      LL = J + NWRDS*(LFT - 1)                                          INARC
      YA = T(LL)                                                        INARC
      IF(KERR .NE. 0) GO TO 20                                          INARC
      DELY = T(KK) - YA                                                 INARC
      Y(I) = YA + FCTR*DELY                                             INARC
   20 CONTINUE                                                          INARC
      WRITE(12)(Y(IJ), IJ = 1, NN)                                      INARC
   30 X = X + H                                                         INARC
      TSTART = TSTART + TAU                                             INARC
   40 CONTINUE                                                          INARC
      RETURN                                                            INARC
  101 CALL ERROR(XINARC, -1, 1)                                         INARC
      RETURN                                                            INARC
      END                                                               INARC
```

1106

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DELT		I	A twenty word array containing the quasitime compute interval for each subarc.	/D	/(211)	CHECK	O	DELT
						ERROR	I	DELT
						INARC	I	DELT
H	h	M	Integration step size in quasitime.	/D	/(2)	AL4	I	H
						INARC	M	H
						MADAMS	I	H
						RKUTT1	I	H
						RKUTT2	I	H
						SALVE	M	H
						WRAPUP	M	DT
IARC	I	M	Subarc number.	/CNTRL	/(24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IDP		I	Component number that corresponds to the QL state variable. IDP = 8.	/PC	/(4)	INARC	I	IDP
						WRAPUP	I	IDP
INARK		I	Logical unit on which initial and converged arcs are stored. INARK = 11.	/GLOBAL	/(95)	CHECK	O	INARK
						FETCH	I	INARK
						INARC	I	INARK
						MARCH	I	INARK
						WRAPUP	I	INARK
MAP		I	An array that maps the initial arc state and costate into the QL state and costate.	/MAP	/(1)	CHECK	O	MAP
						INARC	I	MAP
MOM		M	The number of homogeneous solutions currently being integrated.	/CNTRL	/(9)	GROPE	O	MOM
						INARC	M	MOM
						LINDRV	I	MOM
						NORMAL	I	MOM
						SALVE	M	MOM
						WRAPUP	M	MOM
N		I	Total number of QL state and costate variables. N = 18.	/PC	/(2)	BNDRY	I	N
						CHECK	I	N
						INARC	I	N
						LINDRV	I	N
						NLDRV	I	N
						NORMAL	I	N
						RKUTT1	I	N
						SALVE	I	N
						WRAPUP	I	N
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL	/(18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRO	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						QLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NN		M	The number of quantities currently being numerically integrated.	/CNTRL /(	52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NORMAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAQ	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NPOINT		O	A twenty word array containing the number of points in each subarc.	/D /(	191)	INARC	O	NPOINT
						SALVE	I	NPOINT
NPTS		M	The total number of points in the subarc.	/CNTRL /(	19)	BCOND	O	NPTS
						BNBRY	O	NPTS
						FORCES	I	NPTS
						INARC	M	NPTS
						MAGIC	O	NPTS
						SALVE	M	NPTS
						WRAPUP	O	NPTS
NUP		I	Same as NU.	/CNTRL /(	23)	CHECK	O	NUP
						GROPE	I	NUP
						INARC	I	NUP
QT		O	A twenty word array containing the values from the initial arc of the successive subarcs' durations	/D /(	171)	INARC	O	QT
						WRAPUP	M	QT
TAU	τ	M	Subarc duration (SEC)	/D /(	98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU
TO	t_0	M	Trajectory start time. (SEC)	/GLOBAL/(	7)	FETCH	M	TO
						INARC	M	TO
						TRAJIN	I	TO
						WRAPUP	I	TO
X	x	M	The quasitime variable.	/D /(	1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
Y		M	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y /(	1)	GROPE	O	Y
						INARC	M	Y
						MADAMS	M	Y
						QLTOSZ	I	Y
						RKUTT1	M	Y
						SALVE	M	Y
						WRAPUP	I	Y

SUBRØUTINE INTERP

Purpose

INTERP interpolates the total solution from the preceding QL iteration at the midpoints of the first three compute intervals of each subarc. The results are used by the Runge-Kutta starting procedure.

```
      SUBROUTINE INTERP                                                 INTERP
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QI(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
C                                                                       INTERP
C     THIS SUBROUTINE INTERPOLATES THE NOMINAL SOLUTION AT THE MID-     INTERP
C     POINTS OF THE FIRST THREE COMPUTE INTERVALS.  A FOURTH-ORDER AIT- INTERP
C     KENS ALGORITHM IS USED.                                           INTERP
C                                                                       INTERP
      COMMON /Z/ Z(50)                                                  Z
      COMMON /ZI/ ZI(20, 4)                                             ZI
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      DIMENSION P(20, 4)                                                INTERP
C                                                                       INTERP
      NEWNOM = .TRUE.                                                   INTERP
      DO 1 J = 1, 4                                                     INTERP
      DO 1 I = 1, NNP                                                   INTERP
    1 P(I, J) = ZI(I, J)                                                INTERP
      DO 2 K = 1, 3                                                     INTERP
      S = XI(K) - X                                                     INTERP
      L = K + 1                                                         INTERP
      DO 2 J = L, 4                                                     INTERP
      R = XI(J) - X                                                     INTERP
      Q = R - S                                                         INTERP
      U = R/Q                                                           INTERP
      V = S/Q                                                           INTERP
      DO 2 I = 1, NNP                                                   INTERP
    2 P(I,J) = U*P(I, K) - V*P(I, J)                                    INTERP
      DO 3 I = 1, NNP                                                   INTERP
    3 Z(I) = P(I, 4)                                                    INTERP
      RETURN                                                            INTERP
      END                                                               INTERP
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NEWNOM		O	A logical flag that indicates to the Runge-Kutta integration whether or not the system Jacobian needs to be reevaluated.	/CNTRL	/(15)	INTERP	O	NEWNOM
						LINDRV	M	NEWNOM
						RKUTT1	O	NEWNOM
						SALVE	O	NEWNOM
						WRAPUP	O	NEWNOM
NNP		I	Number of QL state and costate variables.(18)	/CNTRL	/(22)	CHECK	M	NNP
						INTERP	I	NNP
V	V	M	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
X	x	I	The quasitime variable.	/D	/(1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
XI		I	A four word array containing the first four values of quasitime in the subarc.	/D	/(3)	INTERP	I	XI
						SALVE	O	XI
Z	Z	O	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRO	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZI		I	A 20x4 array containing the first four values of Z in the present subarc.	/ZI	/(1)	INTERP	I	ZI
						RKUTT1	I	ZI
						SALVE	M	ZI

SUBRØUTINE
INTRPT

1113

Purpose

INTRPT evaluates the state and costate target misses at an intermediate point and their partials with respect to the c's.*

---

*See Sections 16.6 and 17.4 of Vol. I.

INTRPT

```
          SUBROUTINE INTRPT(DPZIDC, KK)                                    INTRPT
    C                                                                      INTRPT
    C                   THIS ROUTINE EVALUATES THE STATE AND COSTATE TARGET  INTRPT
    C                   COND. MISSES AT AN INTERMEDIATE CORNER PT.  IT ALSO  INTRPT
    C                   EVALUATES THE PARTS.W/RESP.TO THE C*S OF THE MISSES. INTRPT
    C                                                                      INTRPT
         .REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM  D
         *, LHT                                                            D
          COMMON /D/                                                       D
         *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI, D
         *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,  JUL21
         *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20) D
          DIMENSION NOM(20)                                                D
          EQUIVALENCE (NOM, V)                                             D
          COMMON /CNTRL/                                                   CNTRL
         *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT  .,MOM  ,   CNTRL
         *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,  CNTRL
         *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,  CNTRL
         *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES         CNTRL
          LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                   CNTRL
          COMMON/GLOBAL/                                                   GLOBAL
         *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,  GLOBAL
         *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
         *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,      GLOBAL
         *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                         GLOBAL
          COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20), BLOCK
         *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),          BLOCK
         *VALTC(10, 20), IPAY                                              BLOCK
          COMMON /EVAL/ SGN, SPART(18), MAP(10), PZI(40), NOCK, S(18, 41), EVAL
         *TEMP(40), DZ(18), DC, L, SI(18, 41)                              EVAL
          COMMON /Z/ Z(50)                                                 Z
          DIMENSION AI(18, 18), BI(18), DPZIDC(KK, KK), VAL(18), ZZ(18),   INTRPT
         *TOM(18)                                                          INTRPT
          DO 101 I = 1, 18                                                 INTRPT
          DO 101 J = 1, 18                                                 INTRPT
      101 AI(I, J) = 0                                                     INTRPT
          MA = 0                                                           INTRPT
          MTARG = JTAB(IARC)                                               INTRPT
    C                   STORE THE NUMBER OF STATE AND COSTATE TARGET COND. INTRPT
    C                   AT THIS PT.                                        INTRPT
          NTARG = LTAB(IARC)                                               INTRPT
          ASSIGN 109 TO LABL                                               INTRPT
    C                   ARE THERE ANY COSTATE TARGETS.                     INTRPT
          IF(NTARG  LE. 0) GO TO 109                                       INTRPT
    C                   YES.  SET UP THOSE ROWS OF AI MATRIX THAT RESULT   INTRPT
    C                   FROM THE INITIAL CONDS. ON STATE.                  INTRPT
      102 DO 108 I = 1, 9                                                  INTRPT
          IF(IIC(I, IARC + 1) - 1) 103, 104, 105                           INTRPT
    C                   CONTINUOUS STATE                                   INTRPT
      103 MA = MA + 1                                                      INTRPT
          AI(MA, I) = 1                                                    INTRPT
          AI(MA, I + 9) = -1                                               INTRPT
          GO TO 108                                                        INTRPT
    C                   KNOWN STATE                                        INTRPT
      104 MA = MA + 1                                                      INTRPT
          AI(MA, I) = 1                                                    INTRPT
          GO TO 108                                                        INTRPT
      105 IF(IIC(I, IARC + 1) - 5) 108, 106, 107                           INTRPT
    C                   KNOWN DROP WEIGHT                                  INTRPT
      106 MA = MA + 1                                                      INTRPT
          AI(MA, I) = -1                                                   INTRPT
          AI(MA, I + 9) = 1                                                INTRPT
          GO TO 108                                                        INTRPT
      107 WPRO = GR*(ZSAVE(7) - M)                                         INTRPT
    C                   SIZING DROP WEIGHT                                 INTRPT
          CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                 INTRPT
          AI(MA, I) = -1                                                   INTRPT
          AI(MA, I + 9) = 1. - DWDRP                                       INTRPT
      108 CONTINUE                                                         INTRPT
          GO TO LABL                                                       INTRPT
    C                   ARE THERE ANY STATE TARGET CONDITIONS              INTRPT
```

Sizing Interface (SIZIN) 7

```
      109 IF(MTARG .LE. 0) GO TO 115                                    INTRPT
C                     YES.  ADD THOSE ROWS TO AI MATRIX THAT RESULT FROM  INTRPT
C                     STATE TARGET CONDS.                                 INTRPT
          DO 114 I = 1, MTARG                                           INTRPT
          MA = MA + 1                                                   INTRPT
          K = L + I                                                     INTRPT
          KODE = ITC(I, IARC)                                           INTRPT
          IF(KODE .GT. 11) GO TO 111                                    INTRPT
C                     SIMPLE RELATIVE STATE MATCH                         INTRPT
          KODE = MAP(KODE)                                              INTRPT
          AI(MA, KODE + 9) = 1                                          INTRPT
C                     COMP. AND STORE TARGET MISS                         INTRPT
          PZI(K) = NOM(KODE) - VALTC(I, IARC)                           INTRPT
C                     STORE PARTIALS OF MISS                              INTRPT
          DO 110 J = 1, NOCK                                            INTRPT
      110 DPZIDC(K, J) = S(KODE, J + 1)                                 INTRPT
          GO TO 114                                                     INTRPT
C                     COMPLEX CONDITION ON STATE                          INTRPT
      111 ISKIP = 0                                                     INTRPT
          CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                INTRPT
          DO 112 J = 1, 9                                               INTRPT
      112 AI(MA, J + 9) = SPART(J)                                      INTRPT
C                     COMP. AND STORE TARGET MISS                         INTRPT
          PZI(K) = VAL - VALTC(I, IARC)                                 INTRPT
C                     COMP. AND STORE PARTIALS OF TARGET MISS             INTRPT
          CALL MATMLT(TEMP, SPART, S(1, 2), 1, 18, NOCK)                INTRPT
          DO 113 J = 1, NOCK                                            INTRPT
      113 DPZIDC(K, J) = TEMP(J)                                        INTRPT
      114 CONTINUE                                                      INTRPT
C                     ARE THERE ANY COSTATE TARGETS                       INTRPT
      115 IF(NTARG .LE. 0) GO TO 129                                    INTRPT
C                     YES.  COMPLETE AND INVERT THE AI MATRIX             INTRPT
          CALL BASIS(AI, MA, 18)                                        INTRPT
          CALL GJRV(AI, 18, 1.E-12, IERR)                               INTRPT
C                     SET UP THE BI VECTOR.                               INTRPT
          KNOCK = NOC(IARC + 1)                                         INTRPT
          CALL READMS(41, SI, 18*(KNOCK + 1), 2*IARC + 1)               INTRPT
          CALL MATMLT(ZZ, SI(1, 2), C, 18, KNOCK, 1)                    INTRPT
          CALL MATADD(ZZ , ZZ, DZ, 18, 1)                               INTRPT
          DO 116 I = 10, 18                                             INTRPT
          BI(I - 9) = -ZZ(I)                                            INTRPT
      116 BI(I) = NOM(I)                                                INTRPT
          IF(IPAV.EQ.9.AND..NOT.(NBRAN.LT.IARC.AND.IARC.LT.NFARC))      SEP15
         *BI(17) = BI(17) + 1.                                          SEP15
          IF = MA + 1                                                   INTRPT
          IL = 18 - MA                                                  INTRPT
          IP = L + MTARG                                                INTRPT
C                     EVALUATE THE TRANS. CONDS.                          INTRPT
          CALL MATMLT(VAL, BI, AI(1, IF), 1, 18, IL)                    INTRPT
C                     STORE NON-TRIVIAL TRANS.CONDS. AS COSTATE TARGET    INTRPT
C                     MISSES                                              INTRPT
          DO 117 I = 1, NTARG                                           INTRPT
          K = IP + I                                                    INTRPT
          J = ITCT(I, IARC)                                             INTRPT
      11, PZI(K) = VAL(J)                                               INTRPT
C                     COMPUTE DIVIDED DIFF. PARTS.W/RESP.TO THE C*S OF    INTRPT
C                     THE COSTATE TARGET MISSES                           INTRPT
          DO 128 IC = 1, KNOCK                                          INTRPT
C                     COMP. STATE/COSTATE PERT. THAT RESULTS FROM THE     INTRPT
C                     PERT.OF THIS C. LATE SIDE                           INTRPT
          CALL MATMLT(DZ, SI(1, IC + 1), DC, 18, 1, 1)                  INTRPT
C                     ADD STATE/COSTATE PERT. TO BASE VALUE. LATE SIDE.   INTRPT
          CALL MATADD(TOM, ZZ, DZ, 18, 1)                               INTRPT
          IF(IC - NOCK - 1) 118, 119, 1201                              INTRPT
C                     THIS C HAS AN EFFECT ON THE EARLY SIDE, SO COMP.    INTRPT
C                     ITS EFFECT ON THE EARLY SIDE STATE/COSTATE VECTOR   INTRPT
      118 CALL MATMLT(DZ, S(1, IC + 1), DC, 18, 1, 1)                   INTRPT
          CALL MATADD(NOM, Z, DZ, 18, 1)                                INTRPT
          GO TO 120                                                     INTRPT
C                     THIS C AND ALL SUBSEQUENT C*S IN THIS LOOP ARE NOT  INTRPT
C                     TO HAVE AN EFFECT ON THE EARLY SIDE, SO RESTORE     INTRPT
C                     THE EARLY SIDE STATE/COSTATE VECTOR TO ITS BASE VAL. INTRPT
```

```
  119 DO 1191 I = 1, 18                                                      INTRPT
 1191 NOM(I) = Z(I)                                                          INTRPT
  120 CALL FORCES                                                            INTRPT
C                          RECOMPUTE AI MATRIX                               INTRPT
 1201 DO 121 I = 1, 18                                                       INTRPT
      DO 121 J = 1, 18                                                       INTRPT
  121 AI(I, J) = 0                                                           INTRPT
      MA = 0                                                                 INTRPT
      ASSIGN 122 TO LABL                                                     INTRPT
      GO TO 102                                                              INTRPT
  122 IF(MTARG .LE. 0) GO TO 1251                                            INTRPT
      DO 125 I = 1, MTARG                                                    INTRPT
      MA = MA + 1                                                            INTRPT
      KODE = ITC(I, IARC)                                                    INTRPT
      IF(KODE .GT. 11) GO TO 123                                             INTRPT
      KODE = MAP(KODE)                                                       INTRPT
      AI(MA, KODE + 9) = 1                                                   INTRPT
      GO TO 125                                                              INTRPT
  123 ISKIP = 0                                                              INTRPT
      CALL PDBCOL(KODE, VAL, SPART, RUMMY, 1, ISKIP)                         INTRPT
      DO 124 J = 1, 9                                                        INTRPT
  124 AI(MA, J + 9) = SPART(J)                                               INTRPT
  125 CONTINUE                                                               INTRPT
 1251 CALL BASIS(AI, MA, 18)                                                 INTRPT
      CALL GJRV(AI, 18, 1.E-12, IERR)                                        INTRPT
C                          RECOMPUTE BI MATRIX                               INTRPT
      DO 126 I = 10, 18                                                      INTRPT
      BI(I - 9) = -TOM(I)                                                    INTRPT
  126 BI(I) = NOM(I)                                                         INTRPT
      IF(IPAY.EQ.9 AND..NOT.(NBRAN.LT.IARC.AND.IARC.LT.NFARC))               SEP15
     *BI(17) = BI(17) + 1.                                                   SEP15
C                          REEVALUATE TRANS. CONDS.                          INTRPT
      CALL MATMLT(VAL, BI, AI(1, IF), 1, 18, 1L)                             INTRPT
C                          COMP. AND STORE DIVIDED DIFF. PARTS.W/RESP TO THIS INTRPT
C                          C OF THE COSTATE TARGET MISS.                     INTRPT
      DO 127 I = 1, NTARG                                                    INTRPT
      K = IP + I                                                             INTRPT
      J = ITCT(I, IARC)                                                      INTRPT
  127 DPZIDC(K, IC) = (VAL(J) - PZI(K))/DC                                   INTRPT
  12  CONTINUE                                                               INTRPT
  129 L = L + MTARG + NTARG                                                  INTRPT
C                          UPDATE NUMBER OF STATE AND COSTATE TARGET MISSES  INTRPT
C                          COMPUTED SO FAR                                   INTRPT
      RETURN                                                                 INTRPT
      END                                                                    INTRPT
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D /(	11)	BNDRY BRANPT GROPE INTRPT NEWCS NLDRV NORMAL WRAPUP	I I I I M I I I	C C C C C C C C
DC	Δc_i	I	Small perturbation of a c.	/EVAL /(	867)	BNDRY BRANPT ENDPT INTRPT	O I I I	DC DC DC DC
DZ	$\Delta c_i h_i(I^-)$	I	An 18 word array that contains the second term on the right hand side of Equation 17.4-11 of Vol.I of this document.	/EVAL /(	849)	BRANPT ENDPT INTRPT	I I I	DZ DZ DZ
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5 APPLY BRANPT COSTAB COSTAI INTRPT OUTPUT PDBCOL QLTOSZ SALVE STATEF TH3	I I I I I I I I I I I I	GR GR GR GR GR GR GR GR GR GR GR GR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN BCOND BNDRY BRANPT CHECK COSTAB COSTAI ENDPT FORCES INARC INTRPT MAGIC MARCH QLTOSZ SALVE WRAPUP	I M M I M I I I I M I M I I M M	IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND BRANPT CHECK COSTAB COSTAI COSTAO INTRPT SALVE	M I I I I I I I	IIC IIC IIC IIC IIC IIC IIC IIC
ITC		I	A 10x20 array containing the initial condition codes for the QL costate vector. The columns correspond to subarc starting points, the rows, to QL costate variables.	/BLOCK /(	401)	BCOND BRANPT CHECK COSTAB COSTAI ENDPT INTRPT	O I I I I I I	ITC ITC ITC ITC ITC ITC ITC
ITCT		I	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND BRANPT CHECK COSTAB COSTAI INTRPT MAGIC	O I I O O I O	ITCT ITCT ITCT ITCT ITCT ITCT ITCT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB
L		M	Total number of target conditions to satisfy in the problem.	/EVAL /(	868)	BNDRY	M	L
						BRANPT	M	L
						ENDPT	M	L
						INTRPT	M	L
LTAB		I	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND	O	LTAB
						BRANPT	I	LTAB
						COSTAB	O	LTAB
						COSTAI	O	LTAB
						INTRPT	I	LTAB
						MAGIC	M	LTAB
M	m	I	Mass (G'S)	/D /(	97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
MAP		I	A 10 word array that maps the steepest descent state vector into the QL state vector.	/EVAL /(	20)	BNDRY	O	MAP
						BRANPT	I	MAP
						ENDPT	I	MAP
						INTRPT	I	MAP
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY	I	NBRAN
						BRANPT	I	NBRAN
						COSTAB	I	NBRAN
						ENVPRO	I	NBRAN
						INTRPT	I	NBRAN
						MAGIC	I	NBRAN
						QLTOSZ	I	NBRAN
						SALVE	I	NBRAN
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRO	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						QLTOSZ	I	NFARC
						SALVE	I	NFARC
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAO	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOCK	m_I	I	The number of c's in the vector C_1 defined by Equation 17.4-4 of Vol.I of this document.	/EVAL /(	70)	BNDRY	M	NOCK
						BRANPT	I	NOCK
						ENDPT	I	NOCK
						INTRPT	I	NOCK

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NOM	V	M	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRQ	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCQL	I	V
						STATEF	I	V
						WRAPUP	I	V
PZI		M	A 40 word array that contains the target condition misses for all the target conditions in the problem.	/EVAL	/(30)	BNDRY	I	PZI
						BRANPT	M	PZI
						ENDPT	M	PZI
						INTRPT	M	PZI
S		I	An 18x41 array used to store the particular and homogeneous solutions on the early side of a corner point.	/EVAL	/(71)	BNDRY	I	S
						BRANPT	I	S
						ENDPT	I	S
						INTRPT	I	S
SI		I	An 18x41 array used to store the particular and homogeneous solutions on the late side of a corner point.	/EVAL	/(869)	BRANPT	I	SI
						INTRPT	I	SI
SPART		I	An 18 word array whose first nine entries receive the values of the partial derivatives wrt the state of those target conditions computed in subroutine PDBCQL.	/EVAL	/(2)	BNDRY	O	SPART
						BRANPT	I	SPART
						ENDPT	I	SPART
						INTRPT	I	SPART
TEMP	$(\partial\psi_I/\partial C_I)^T$	I	A 40 word array that contains the transpose of the vector defined by Equation 17.4-9 of Vol.I of this document.	/EVAL	/(809)	BRANPT	I	TEMP
						ENDPT	I	TEMP
						INTRPT	I	TEMP
VALTC		I	A 10x20 array containing the desired values of the state target conditions whose codes appear in the array IICT.	/BLOCK	/(1062)	BCOND	O	VALTC
						BRANPT	I	VALTC
						CHECK	I	VALTC
						ENDPT	I	VALTC
						INTRPT	I	VALTC
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRQ	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D	/(151)	BCOND	O	ZSAVE
						BRANPT	I	ZSAVE
						COSTAB	I	ZSAVE
						COSTAI	I	ZSAVE
						INTRPT	I	ZSAVE
						PDBCQL	I	ZSAVE
						SALVE	I	ZSAVE

1120

SUBRØUTINE LINDRV

Purpose

LINDRV computes the quasitime derivatives of the particular and homogeneous solutions. These derivatives are defined by Equations 17.1-9 and 17.1-10 in Vol. I.

```
      SUBROUTINE LINDRV(Y, F)                                           LINDRV
C                                                                       LINDRV
C                   THIS ROUTINE COMPUTES THE DERIVATIVES OF THE PARTI- LINDRV
C                   CULAR AND HOMOGENEOUS SOLUTIONS                     LINDRV
C                                                                       LINDRV
      COMMON /JACOB/ JAKE(400)                                          JACOB
      COMMON /Z/ Z(50)                                                  Z
      COMMON /ZD/ ZD(50)                                                ZD
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      DIMENSION P(20), Y(1), F(1)                                       LINDRV
      IF(NEWNOM) CALL NLDRV(Z, ZD)                                      LINDRV
      NEWNOM = .FALSE.                                                  LINDRV
      NP1 = N + 1                                                       LINDRV
      CALL MATMLT(F(NP1), JAKE, Y(NP1), N, N, MOM)                      LINDRV
      CALL MATSUB(P, Y, Z, N, 1)                                        LINDRV
      CALL MATMLT(F, JAKE, P, N, N, 1)                                  LINDRV
      CALL MATADD(F, F, ZD, N, 1)                                       LINDRV
      RETURN                                                            LINDRV
      END                                                               LINDRV
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
JAKE		I	An 18x18 array defined by Equation 17.5-5 in Vol.I of this document. The entry in the i-th row and j-th column is the total partial derivative of the quasitime derivative of the Y_i component of Y with respect to the Y_j component of Y, i.e., $\partial\dot{Y}_i/\partial Y_j$, where $Y^T = (y^T, \lambda^T)$	/JACOB /(	1)	LINDRV NLDRV NLDRV SALVE	I M M O	JAKE JACOB VDV JAKE
MOM		I	The number of homogeneous solutions currently being integrated.	/CNTRL /(	9)	GROPE INARC LINDRV NOMNAL SALVE WRAPUP	O M I I M M	MOM MOM MOM MOM MOM MOM
N		I	Total number of QL state and costate variables. N = 18.	/PC /(	2)	BNDRV CHECK INARC LINDRV NLDRV NOMNAL RKUTT1 SALVE WRAPUP	I I I I I I I I I	N N N N N N N N N
NEWNOM		M	A logical flag that indicates to the Runge-Kutta integration whether or not the system Jacobian needs to be reevaluated.	/CNTRL /(	15)	INTERP LINDRV RKUTT1 SALVE WRAPUP	O M O O O	NEWNOM NEWNOM NEWNOM NEWNOM NEWNOM
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z /(	1)	BNDRV BRANPT ENDPT ENVPRQ INTERP INTRPT LINDRV NOMNAL OUTPUT RKUTT1 RKUTT2 SALVE WRAPUP	I I I I O I I M I O M M M	Z Z Z Z Z Z Z Z Z Z Z Z Z
ZD		I	A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.I of this document.	/ZD /(	1)	ENVPRQ LINDRV OUTPUT RKUTT2 WRAPUP	I I I I I	ZD ZD ZD ZD ZD

SUBRØUTINE
MADAMS

Purpose

MADAMS carries out the fourth order Adams-Moulton integration of the particular and homogeneous solutions.*

*See Section 17.6 of Vol. I.

```
      SUBROUTINE MADAMS                                                 MADAMS
C                                                                       MADAMS
C                    THIS ROUTINE CARRIES OUT THE FOURTH ORDER ADAMS-   MADAMS
C                    MOULTON INTEGRATION OF THE PARTICULAR AND HOMO-    MADAMS
C                    GENEOUS SOLUTIONS                                  MADAMS
C                                                                       MADAMS
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,        GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /Y/ Y(820)                                                 Y
      COMMON /F/ F(820, 4)                                              F
      COMMON /STUFF/ F1(820), F2(820), YP(820), YC(820)                 MADAMS
      DIMENSIONCC(5), P(4)                                              MADAMS
      DATA CC /-19., 106., -264., 646., 251./,                          MADAMS
     *P(1), P(2), P(3), P(4)/ -9., 37., -59., 55./                      MADAMS
C                                                                       MADAMS
C                    SET SCALING FACTORS AND INITIALIZE PREDICTOR AND   MADAMS
C                    CORRECTOR                                          MADAMS
      HP = H/24.                                                        MADAMS
      HC = H/720.                                                       MADAMS
      DO 1 I = 1, NN                                                    MADAMS
      YP(I) = Y(I)                                                      MADAMS
    1 YC(I) = Y(I)                                                      MADAMS
C                    ACCUMULATE FIRST FOUR TERMS OF PRED. AND CORR.     MADAMS
      DO 2 K = 1, 4                                                     MADAMS
      L = INDX(K)                                                       MADAMS
      TP = HP*P(K)                                                      MADAMS
      TC = HC*CC(K)                                                     MADAMS
      DO 2 I = 1, NN                                                    MADAMS
      FIL = F(I, L)                                                     MADAMS
      YP(I) = YP(I) + TP*FIL                                            MADAMS
    2 YC(I) = YC(I) + TC*FIL                                            MADAMS
C                    INCREMENT INDEP. VAR. AND ADD LAST TERM TO CORR.   MADAMS
      X = X + H                                                         MADAMS
      CALL LINDRV(YP, F1)                                               MADAMS
      TC = HC*CC(5)                                                     MADAMS
      DO 3 I = 1, NN                                                    MADAMS
    3 YC(I) = YC(I) + TC*F1(I)                                          MADAMS
      J = 1                                                             MADAMS
C                    MORE INNER LOOPS.                                  MADAMS
    4 IF(J .GE. INNER) GO TO 6                                          MADAMS
C                    YES.  MAKE PRED. = CORR. AND UPDATE CORR.          MADAMS
      CALL LINDRV(YC, F2)                                               MADAMS
      DO 5 I = 1, NN                                                    MADAMS
      YP(I) = YC(I)                                                     MADAMS
      YC(I) = YC(I) + TC*(F2(I) - F1(I))                                MADAMS
    5 F1(I) = F2(I)                                                     MADAMS
      J = J + 1                                                         MADAMS
      GO TO 4                                                           MADAMS
C                    ROTATE INDEX POINTERS AND SET INTEGRAL = CORR.     MADAMS
    6 ISAVE = INDX(1)                                                   MADAMS
      INDX(1) = INDX(2)                                                 MADAMS
      INDX(2) = INDX(3)                                                 MADAMS
      INDX(3) = INDX(4)                                                 MADAMS
      INDX(4) = ISAVE                                                   MADAMS
      DO 8 I = 1, NN                                                    MADAMS
    8 Y(I) = YC(I)                                                      MADAMS
      CALL LINDRV(Y, F(1, ISAVE))                                       MADAMS
```

6-

4

1127

```
75.        RETURN                                                      MADAMS
76.        END                                                         MADAMS
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
F		I	An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-7 thru -10 of Vol.I of this document.	/F	/(1)	MADAMS SALVE	I I	F F
F1		M	An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-2 thru -5 in Vol.I of this document.	/STUFF	/(1)	MADAMS RKUTT1 RKUTT1	M I O	F1 FK FS
H	h	I	Integration step size in quasitime.	/D	/(2)	AL4 INARC MADAMS RKUTT1 RKUTT2 SALVE WRAPUP	I M I I I M M	H H H H H H DT
INDX		M	An array of four words that indicate to Adams-Moulton integration in what order the derivatives of the particular and homogeneous solutions are stored.	/CNTRL	/(11)	BCOND MADAMS SALVE	M M O	INDX INDX INDX
INNER		I	Number of Adams-Moulton inner loops.	/GLOBAL	/(9)	CHECK MADAMS	M I	INNER INNER
NN		I	The number of quantities currently being numerically integrated.	/CNTRL	/(52)	BNDRY INARC MADAMS MAGIC NOMNAL RKUTT1 RKUTT2 SALVE WRAPUP	M M I M I I I M M	NN NN NN NN NN NN NN NN NN
X	x	M	The quasitime variable.	/D	/(1)	AL4 BNDRY ERROR FETCH FORCES INARC INTERP MADAMS RKUTT1 RKUTT2 SALVE STATEF WRAPUP	I O I O I M I M M M M I M	X X X X X X X X X X X X TT
Y		M	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y	/(1)	GROPE INARC MADAMS QLTOSZ RKUTT1 SALVE WRAPUP	O M M I M M I	Y Y Y Y Y Y Y

Purpose

MAG computes the magnitude of a vector v drawn from the Euclidean space E^n. This magnitude is defined as

$$||v|| = \sum_{i=1}^{n} |v_i|.$$

```
      FUNCTION MAG(V, N)                                                MAG
      REAL MAG                                                          MAG
C                                                                       MAG
C     THIS SUBPROGRAM COMPUTES THE SCALER MAGNITUDE OF THE VECTOR V.    MAG
C                                                                       MAG
      DIMENSION V(N)                                                    MAG
C                                                                       MAG
      SUM = 0.                                                          MAG
      DO 1 I = 1, N                                                     MAG
    1 SUM = SUM +  ABS(V(I))                                            MAG
      MAG = SUM                                                         MAG
      RETURN                                                            MAG
      END                                                               MAG
```

SUBRØUTINE MAGIC

Purpose

MAGIC controls the determination of the costate initial and target conditions over all of the corner points in the problem.*

---

*See Sections 16.6, 17.1 and 17.2.

```
      SUBROUTINE MAGIC                                                  MAGIC
C                                                                       MAGIC
C                     THIS ROUTINE CONTROLS THE DETERMINATION OF THE    MAGIC
C                     COSTATE INITIAL AND TARGET CONDITIONS.            MAGIC
C                                                                       MAGIC
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      DATA MAGIX /6HMAGIC /                                             MAGIC
C                     INITIALIZE THE IICT AND ITCT BLOCKS               MAGIC
      DO 101 I = 1, 9                                                   MAGIC
      DO 101 J = 1, NARC                                                MAGIC
      ITCT(I, J) = 0                                                    MAGIC
  101 IICT(I, J) = 2                                                    MAGIC
      DO 1011 I = 1, NARC                                               MAGIC
 1011 IICT(8, I) = 1                                                    MAGIC
C                     SET INTEGRATION CONTROL FLAGS SO THAT FORCES WILL MAGIC
C                     WILL CALL ARCEN AND ARCIN.                        MAGIC
      NPTS = 1                                                          MAGIC
      KPT = 1                                                           MAGIC
      DO 104 IARC = 1, NARC                                             MAGIC
C                     CLEAR LTAB ENTRY.                                 MAGIC
      LTAB(IARC) = 0                                                    MAGIC
C                     IS THIS THE FIRST ARC                             MAGIC
      IF(IARC .GT. 1) GO TO 102                                         MAGIC
C                     YES. INITIAL POINT.                               MAGIC
      CALL COSTAO                                                       MAGIC
      GO TO 104                                                         MAGIC
C                     IS THIS THE FIRST ARC OF THE FIRST BRANCH.        MAGIC
  102 IF(IARC .NE. NBRAN + 1) GO TO 103                                 MAGIC
C                     YES. BRANCH POINT.                                MAGIC
      CALL COSTAB                                                       MAGIC
      GO TO 104                                                         MAGIC
C                     IS THIS THE FIRST ARC OF THE SECOND BRANCH        MAGIC
  103 IF(IARC .EQ. NFARC + 1) GO TO 104                                 MAGIC
C                     NO.  INTERMEDIATE POINT.                          MAGIC
      CALL COSTAI                                                       MAGIC
  104 CONTINUE                                                          MAGIC
C                     COMPUTE TOTAL NUMBER OF STATE AND COSTATE TARGET  MAGIC
C                     CONDITIONS TO MATCH                               MAGIC
      NCOND = 9                                                         MAGIC
      IF(NFARC .LT. NARC) NCOND = 18                                    MAGIC
      NN = NARC - 1                                                     MAGIC
      DO 105 IARC = 1, NN                                               MAGIC
      IF(IARC .EQ. NFARC) GO TO 105                                     MAGIC
  105 NCOND = NCOND + JTAB(IARC) + LTAB(IARC)                           MAGIC
C                     MAKE SURE NUMBER OF TARGETS EQUALS NUMBER OF      MAGIC
C                     UNKNOWN INITIAL VALUES.                           MAGIC
      IF(NCOND - NOKNOW) 116, 118, 117                                  MAGIC
  116 CALL ERROR(MAGIX, -1, 1)                                          MAGIC
  117 CALL ERROR(MAGIX, -2, 1)                                          MAGIC
  118 RETURN                                                            MAGIC
      END                                                               MAGIC
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
IARC	I	M	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IICT		O	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK	I	IICT
						COSTAB	M	IICT
						COSTAI	M	IICT
						COSTAO	O	IICT
						MAGIC	O	IICT
						SALVE	I	IICT
ITCT		O	A 10x20 array containing the QL costate analog to the array IICT. The columns correspond to subarc end points. The nonzero entries in a column are the numbers of those components of the vector θ in Equation 16.6-34 of Volume I of the PADS document that contain the value of a costate-target condition that applies at the end of the corresponding subarc.	/BLOCK /(	621)	BCOND	O	ITCT
						BRANPT	I	ITCT
						CHECK	I	ITCT
						COSTAB	O	ITCT
						COSTAI	O	ITCT
						INTRPT	I	ITCT
						MAGIC	O	ITCT
JTAB		I	An array containing the number of nonzero entries in each column of the array IICT.	/BLOCK /(	601)	BCOND	M	JTAB
						BRANPT	I	JTAB
						CHECK	I	JTAB
						COSTAB	I	JTAB
						COSTAI	I	JTAB
						ENDPT	I	JTAB
						INTRPT	I	JTAB
						MAGIC	I	JTAB
KPT		O	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND	O	KPT
						BNDRY	O	KPT
						FORCES	I	KPT
						MAGIC	O	KPT
						RKUTT1	I	KPT
						SALVE	M	KPT
						WRAPUP	M	KPT
LTAB		M	An array containing the number of nonzero entries in each column of the array ITCT.	/BLOCK /(	821)	BCOND	O	LTAB
						BRANPT	I	LTAB
						COSTAB	O	LTAB
						COSTAI	O	LTAB
						INTRPT	I	LTAB
						MAGIC	M	LTAB
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRQ	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						QLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY	I	NBRAN
						BRANPT	I	NBRAN
						COSTAB	I	NBRAN
						ENVPRQ	I	NBRAN
						INTRPT	I	NBRAN
						MAGIC	I	NBRAN
						QLTOSZ	I	NBRAN
						SALVE	I	NBRAN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRD	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						OLTOSZ	I	NFARC
						SALVE	I	NFARC
NN		M	The number of quantities currently being numerically integrated.	/CNTRL /(	52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NOMNAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
NOKNOW		I	The total number of free (unknown) state and costate variables over all the subarcs.	/BLOCK /(	841)	CHECK	I	NOKNOW
						CONOMO	I	NOKNOW
						COSTAB	M	NOKNOW
						COSTAI	M	NOKNOW
						COSTAO	M	NOKNOW
						GROPE	I	NOKNOW
						MAGIC	I	NOKNOW
NPTS		O	The total number of points in the subarc.	/CNTRL /(	19)	BCOND	O	NPTS
						BNDRY	O	NPTS
						FORCES	I	NPTS
						INARC	M	NPTS
						MAGIC	O	NPTS
						SALVE	M	NPTS
						WRAPUP	O	NPTS

SUBRØUTINE
MARCH

Purpose

MARCH finds the closest two points on the initial arc file that bracket the desired interpolation time.

```
      SUBROUTINE MARCH(TIME, IFT, T, NWRDS, IERR)                       MARCH
C                                                                       MARCH
C                 THIS ROUTINE FINDS THE CLOSEST TWO PTS. ON THE        MARCH
C                 INITIAL ARC FILE THAT BRACKET THE DESIRED INTER-      MARCH
C                 POLATION TIME.                                        MARCH
C                                                                       MARCH
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NDVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      EQUIVALENCE(TARC, JARC)                                           MARCH
      DIMENSION T(NWRDS, 2)                                             MARCH
  100 IFT = 3 - IFT                                                     MARCH
      READ(INARK) (T(I, IFT), I = 1, NWRDS)                             MARCH
      IF(EOF,INARK) 103, 101                                            MARCH
  101 TARC = T(1, IFT)                                                  MARCH
      IF(JARC .EQ. IARC) GO TO 102                                      MARCH
      BACKSPACE INARK                                                   MARCH
      BACKSPACE INARK                                                   MARCH
      BACKSPACE INARK                                                   MARCH
      READ(INARK) (T(I, IFT), I = 1, NWRDS)                             MARCH
      IFT = 3 - IFT                                                     MARCH
      RETURN                                                            MARCH
  102 IF(T(2, IFT) .LT. TIME) GO TO 100                                 MARCH
      RETURN                                                            MARCH
  103 IERR = 1                                                          MARCH
      RETURN                                                            MARCH
      END                                                               MARCH
```

101 103 102 100

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
INARK		I	Logical unit on which initial and converged arcs are stored. INARK = 11.	/GLOBAL/(	95)	CHECK	O	INARK
						FETCH	I	INARK
						INARC	I	INARK
						MARCH	I	INARK
						WRAPUP	I	INARK

Purpose

MATADD performs the matrix addition

$$S = A + B,$$

where S, A and B are n x m matrices.

MATADD

```
      SUBROUTINE MATADD(S, A, B, N, M)                                   MATADD
C                                                                        MATADD
C                THIS ROUTINE PERFORMS THE MATRIX ADDITION               MATADD
C     S=A+B, WHERE N IS THE NUMBER OF ROWS IN THE A (B) MATRIX AND M IS  MATADD
C     THE NUMBER OF COLUMNS IN THE A (B) MATRIX.                         MATADD
C                                                                        MATADD
      DIMENSION A(1), B(1), S(1)                                         MATADD
C                                                                        MATADD
      J = N*M                                                            MATADD
      DO 1 I = 1, J                                                      MATADD
    1 S(I) = A(I) + B(I)                                                 MATADD
      RETURN                                                             MATADD
      END                                                                MATADD
```

SUBRØUTINE MØMECØ

Purpose

MOMECO computes the aerodynamic moment coefficient C_m. It also computes the partial derivatives

$$\frac{\partial C_m}{\partial \alpha}, \quad \frac{\partial C_m}{\partial M}, \quad \frac{\partial^2 C_m}{\partial \alpha^2}, \quad \frac{\partial^2 C_m}{\partial M^2} \text{ and } \frac{\partial^2 C_m}{\partial \alpha \partial M}$$

```
      SUBROUTINE MOMECO                                                 MOMECO
C                                                                       MOMECO
C                   THIS ROUTINE COMPUTES THE MOMENT COEFFICIENT AND    MOMECO
C                   ITS PARTIALS                                        MOMECO
C                                                                       MOMECO
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CD0   ,CD0M  ,CL0   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CM0   ,CM0M  ,CM0MM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CD0MM ,CLAMM ,CL0M  ,CL0MM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DATA DEG/57.2957795130823/                                        MOMECO
      CM = CM0 + ALPHA*CMA                                              MOMECO
      CMM = CM0M + ALPHA*CMAM                                           MOMECO
      CMAA = 0.                                                         MOMECO
      CMMM = CM0MM + ALPHA*CMAMM                                        MOMECO
      RETURN                                                            MOMECO
      END                                                               MOMECO
```

1145

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
ALPHA	α	I	Angle of attack (RAD)	/DYNA	/(79)	AEROCO	I	ALPHA
						ALGCON	M	ALPHA
						AL2	I	ALPHA
						ARCIN	M	ALPHA
						CONTRL	M	ALPHA
						ENVPRQ	I	ALPHA
						MOMECO	I	ALPHA
						NPLANE	I	ALPHA
						OUTPUT	I	ALPHA
						TRAJIN	O	ALPHA
						UT	I	ALPHA
						WRAPUP	I	ALPHA
CM	C_m	O	Moment coefficient	/DYNA	/(122)	MOMECO	O	CM
						UT	I	CM
CMA	C_{m_α}	I	Moment coefficient slope (RAD^{-1})	/DYNA	/(123)	MOMECO	I	CMA
						STATEF	M	CMA
						UT	I	CMA
CMAA	$\partial C_{m_\alpha}/\partial\alpha$	O	See symbol	/DYNA	/(125)	MOMECO	O	CMAA
						UT	I	CMAA
CMAM	$\partial C_{m_\alpha}/\partial M$	I	See symbol	/DYNA	/(127)	MOMECO	I	CMAM
						STATEF	M	CMAM
						UT	I	CMAM
CMAMM	$\partial^2 C_{m_\alpha}/\partial M^2$	I	See symbol	/DYNA	/(131)	MOMECO	I	CMAMM
						STATEF	M	CMAMM
CMM	$\partial C_m/\partial M$	O	See symbol	/DYNA	/(124)	MOMECO	O	CMM
						UT	I	CMM
CMMM	$\partial^2 C_m/\partial M^2$	O	See symbol	/DYNA	/(126)	MOMECO	O	CMMM
						UT	I	CMMM
CMO	C_{m_0}	I	Moment coefficient at $\alpha = 0$.	/DYNA	/(128)	MOMECO	I	CMO
						STATEF	I	CMO
CMOM	$\partial C_{m_0}/\partial M$	I	See symbol	/DYNA	/(129)	MOMECO	I	CMOM
						STATEF	I	CMOM
CMOMM	$\partial^2 C_{m_0}/\partial M^2$	I	See symbol	/DYNA	/(130)	MOMECO	I	CMOMM
						STATEF	I	CMOMM

1146

SUBRØUTINE NEWCS

Purpose

NEWCS controls the Newton-Raphson iteration for the c's.*

*See Section 17.4 of Vol. I.

```
      SUBROUTINE NEWCS(DO, K)                                           NEWCS
C                                                                       NEWCS
C              THIS ROUTINE CONTROLS THE DETERMINATION OF THE VALUES    NEWCS
C              OF THE C*S THAT CAUSE ZERO TARGET CONDS. MISS.           NEWCS
C                                                                       NEWCS
      REAL MAG                                                          NEWCS
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      DIMENSION DELC(40), DO(K, K), PZI(40)                             NEWCS
      DATA XNEWCS /6H NEWCS/                                            NEWCS
    1 FORMAT(1H0, 25HSOLUTION FOR C'S FOLLOWS.)                         NEWCS
    2 FORMAT(1H , 1H )                                                  NEWCS
    3 FORMAT(1H , 3HC= , 5E15.7)                                        NEWCS
    4 FORMAT(1H , 3HE= , 5E15.7)                                        NEWCS
C                                                                       NEWCS
      WRITE(6, 1)                                                       NEWCS
C             CLEAR C VECTOR, DIVERGENCE COUNTER AND CONVERGENCE FLAG.  NEWCS
C             SET CURRENT ERROR METRIC TO LARGE NUMBER.                 NEWCS
      DO 101 I = 1, K                                                   NEWCS
  101 C(I) = 0                                                          NEWCS
      IDIV = 0                                                          NEWCS
      DLCBAR = 1.E38                                                    NEWCS
      KON = 0                                                           NEWCS
C             START ITERATION                                           NEWCS
      DO 105 IT = 1, 25                                                 NEWCS
C             CLEAR PARTIAL MATRIX                                      NEWCS
      DO 102 I = 1, K                                                   NEWCS
      DO 102 J = 1, K                                                   NEWCS
  102 DO(I, J) = 0                                                      NEWCS
C             EVAL. TARGET MISSES AND PARTS. W/RESP. TO C*S.            NEWCS
      CALL BNDRY(PZI, DO, K)                                            NEWCS
C             SAVE LAST ERROR METRIC AND COMP. NEW ONE.                 NEWCS
      DLPBAR = DLCBAR                                                   NEWCS
      DLCBAR = MAG(PZI, K)/MAGBV                                        NEWCS
C             ARE WE CONVERGED                                          NEWCS
      IF(DLCBAR .GT. ERR) GO TO 1021                                    NEWCS
C             YES.  SET CONV. FLAG                                      NEWCS
      KON = 1                                                           NEWCS
      GO TO 104                                                         NEWCS
C             DID WE DIVERGE                                            NEWCS
 1021 IF(DLCBAR .LE. DLPBAR) GO TO 103                                  NEWCS
C             YES.  INCREMENT DIVERGENCE COUNTER                        NEWCS
      IDIV = IDIV + 1                                                   NEWCS
C             HAVE WE ACCUMULATED 5 DIVERGENCES                         NEWCS
      IF(IDIV .EQ. 5) CALL ERROR(XNEWCS, -1,0)                          NEWCS
      GO TO 104                                                         NEWCS
C             CONVERGING.  DECREMENT DIVERGENCE COUNTER                 NEWCS
  103 IDIV = MAX0(0, IDIV - 1)                                          NEWCS
C             INVERT PARTIALS MATRIX                                    NEWCS
  104 CALL GJRV(DO, K, 1.E-12, IERR)                                    NEWCS
C             MAKE SURE INVERSION OK.                                   NEWCS
      IF(IERR .NE. 0) CALL ERROR(XNEWCS, -2, 1)                         NEWCS
C                    UPDATE C VECTOR                                    NEWCS
      CALL MATMLT(DELC, DO, PZI, K, K, 1)                               NEWCS
      CALL MATSUB(C, C, DELC, K, 1)                                     NEWCS
      IF(KON .NE. 0) GO TO 110                                          NEWCS
  105 CONTINUE                                                          NEWCS
C                    RAN OUT OF ITERATIONS.  SHALL WE CALL IT CONVERGED NEWCS
C                    ANYWAY.                                            NEWCS
      IF(IDIV .NE. 0) CALL ERROR(XNEWCS, -3, 0)                         NEWCS
  110 WRITE(6, 2)                                                       NEWCS
```

1149

```
      WRITE(6, 3) (C(IJ), IJ = 1, K)                                   NEWCS
      WRITE(6,4) (PZI(IJ), IJ = 1, K)                                  NEWCS
      RETURN                                                           NEWCS
      END                                                              NEWCS
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USA SUBR	CODE	VA
C	c	M	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D	/(11)	BNDRY	I	C
						BRANPT	I	C
						GROPE	I	C
						INTRPT	I	C
						NEWCS	M	C
						NLDRV	I	C
						NOMNAL	I	C
						WRAPUP	I	C
ERR		I	Convergence criterion of iteration for the c's.	/D	/(8)	CHECK	O	ERR
						NEWCS	I	ERR
MAGBV		I	The magnitude of all of the desired values of the state target conditions.	/D	/(7)	CHECK	M	MAGB
						NEWCS	I	MAGB

SUBRØUTINE
NLDRV

Purpose

NLDRV evaluates the nonlinear state and costate quasitime derivatives. In addition, if the QL iteration is not converged, it also evaluates the system Jacobian.*

*See Sections 16.6 and 17.5 of Vol. I.

```
      SUBROUTINE NLDRV(Z, ZD)                                               NLDRV
C                                                                           NLDRV
C                  THIS ROUTINE COMPUTES THE NONLINEAR STATE AND CO-        NLDRV
C                  STATE DERIVATIVES.  IF QL IS NOT CONVERGED, IT ALSO      NLDRV
C                  COMPUTES THE PARTIALS OF THESE EQUATIONS W/RESP. TO      NLDRV
C                  THE STATE AND COSTATE, I.E. THE JACOBIAN.                NLDRV
C                                                                           NLDRV
      COMMON/ARCDAT/                                                        ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      ,    ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      ,    ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      ,    ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      ,    ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      ,    ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        ,    ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE           ARCDAT
      DIMENSION  ARCDA(40)                                                  ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                               ARCDAT
C                                                                           NLDRV
      REAL JACOB,                                                           NLDRV
     *      LOTR,  LVTAU,  LMTAU,   LURC,  LURCT, LUTVRC,   LGTV,           NLDRV
     *    LPTCGV,  LPTCG,  LOVTR,                                           NLDRV
     *       MDV,    MDG,    MDP,    MDR,    MDO,    MDM,    MDT,           NLDRV
     *       LVD,    LGD,    LPD,    LRD,    LOD,    LMD,    LTD,           NLDRV
     *        MD,   LVDG,   LVDP,   LVDR,   LVDO,   LVDT,   LGDP,           NLDRV
     *      LGDR,   LGDO,   LGDT,   LPDP,   LPDR,   LPDO,   LPDT,           NLDRV
     *      LRDR,   LRDO,   LRDT,   LODO,   LODT,   LVDV,   LVDM,           NLDRV
     *      LGDM,   LPDM,   LRDM,   LODM,   LGDG,   LMDM,   LMDT,           NLDRV
     *      LTDT,   MDLV,  LVDLV,  LGDLV,  LRDLV,  LMDLV,  LTDLV,           NLDRV
     *              MDLG,  LVDLG,  LGDLG,  LRDLG,  LMDLG,  LTDLG,           NLDRV
     *              MDLP,  LVDLP,  LGDLP,  LRDLP,  LMDLP,  LTDLP,           NLDRV
     *              MDLM,  LVDLM,  LGDLM,  LRDLM,  LMDLM,  LTDLM            NLDRV
      REAL                                                                  NLDRV
     *LGDV,LPDV,LPDG,LRDV,LRDG,LRDP,LODV,LODG,LODP,LODR,LMDV,LMDG,LMDR,     NLDRV
     *LTDV,LTDG,LTDP,LTDR,LTDO,LTDM,LVDLH,LRDLH,LTDLH,                      NLDRV
     *LPDLV,LODLV,LPDLG,LODLG,LPDLP,LODLP,LVDLR,LGDLR,LTDLR,LVDLO,LGDLO,    NLDRV
     *LPDLO,LRDLO,LTDLO,LVDLU,LGDLU,LPDLU,LRDLU,LODLU,LTDLU                 NLDRV
      DIMENSION Z(16),ZD(16)                                                NLDRV
      COMMON /JACOB/                                                        NLDRV
     *VDV    ,GDV    ,PDV    ,RDV    ,ODV    ,UDV    ,MDV    ,TDV    ,HTDV   ,   NLDRV
     *LVDV   ,LGDV   ,LPDV   ,LRDV   ,LODV   ,LUDV   ,LMDV   ,LTDV   ,LHTDV  ,   NLDRV
     *VDG    ,GDG    ,PDG    ,RDG    ,ODG    ,UDG    ,MDG    ,TDG    ,HTDG   ,   NLDRV
     *LVDG   ,LGDG   ,LPDG   ,LRDG   ,LODG   ,LUDG   ,LMDG   ,LTDG   ,LHTDG  ,   NLDRV
     *VDP    ,GDP    ,PDP    ,RDP    ,ODP    ,UDP    ,MDP    ,TDP    ,HTDP   ,   NLDRV
     *LVDP   ,LGDP   ,LPDP   ,LRDP   ,LODP   ,LUDP   ,LMDP   ,LTDP   ,LHTDP  ,   NLDRV
     *VDR    ,GDR    ,PDR    ,RDR    ,ODR    ,UDR    ,MDR    ,TDR    ,HTDR   ,   NLDRV
     *LVDR   ,LGDR   ,LPDR   ,LRDR   ,LODR   ,LUDR   ,LMDR   ,LTDR   ,LHTDR  ,   NLDRV
     *VDO    ,GDO    ,PDO    ,RDO    ,ODO    ,UDO    ,MDO    ,TDO    ,HTDO   ,   NLDRV
     *LVDO   ,LGDO   ,LPDO   ,LRDO   ,LODO   ,LUDO   ,LMDO   ,LTDO   ,LHTDO  ,   NLDRV
     *VDU    ,GDU    ,PDU    ,RDU    ,ODU    ,UDU    ,MDU    ,TDU    ,HTDU   ,   NLDRV
     *LVDU   ,LGDU   ,LPDU   ,LRDU   ,LODU   ,LUDU   ,LMDU   ,LTDU   ,LHTDU  ,   NLDRV
     *VDM    ,GDM    ,PDM    ,RDM    ,ODM    ,UDM    ,MDM    ,TDM    ,HTDM   ,   NLDRV
     *LVDM   ,LGDM   ,LPDM   ,LRDM   ,LODM   ,LUDM   ,LMDM   ,LTDM   ,LHTDM  ,   NLDRV
     *VDT    ,GDT    ,PDT    ,RDT    ,ODT    ,UDT    ,MDT    ,TDT    ,HTDT   ,   NLDRV
     *LVDT   ,LGDT   ,LPDT   ,LRDT   ,LODT   ,LUDT   ,LMDT   ,LTDT   ,LHTDT  ,   NLDRV
     *VDH    ,GDH    ,PDH    ,RDH    ,ODH    ,UDH    ,MDH    ,TDH    ,HTDH   ,   NLDRV
     *LVDH   ,LGDH   ,LPDH   ,LRDH   ,LODH   ,LUDH   ,LMDH   ,LTDH   ,LHTDH      NLDRV
      COMMON /JACOB/                                                        NLDRV
     *VDLV   ,GDLV   ,PDLV   ,RDLV   ,ODLV   ,UDLV   ,MDLV   ,TDLV   ,HTDLV  ,   NLDRV
     *LVDLV  ,LGDLV  ,LPDLV  ,LRDLV  ,LODLV  ,LUDLV  ,LMDLV  ,LTDLV  ,LHTDLV ,   NLDRV
     *VDLG   ,GDLG   ,PDLG   ,RDLG   ,ODLG   ,UDLG   ,MDLG   ,TDLG   ,HTDLG  ,   NLDRV
     *LVDLG  ,LGDLG  ,LPDLG  ,LRDLG  ,LODLG  ,LUDLG  ,LMDLG  ,LTDLG  ,LHTDLG ,   NLDRV
     *VDLP   ,GDLP   ,PDLP   ,RDLP   ,ODLP   ,UDLP   ,MDLP   ,TDLP   ,HTDLP  ,   NLDRV
     *LVDLP  ,LGDLP  ,LPDLP  ,LRDLP  ,LODLP  ,LUDLP  ,LMDLP  ,LTDLP  ,LHTDLP ,   NLDRV
     *VDLR   ,GDLR   ,PDLR   ,RDLR   ,ODLR   ,UDLR   ,MDLR   ,TDLR   ,HTDLR  ,   NLDRV
     *LVDLR  ,LGDLR  ,LPDLR  ,LRDLR  ,LODLR  ,LUDLR  ,LMDLR  ,LTDLR  ,LHTDLR ,   NLDRV
     *VDLO   ,GDLO   ,PDLO   ,RDLO   ,ODLO   ,UDLO   ,MDLO   ,TDLO   ,HTDLO  ,   NLDRV
     *LVDLO  ,LGDLO  ,LPDLO  ,LRDLO  ,LODLO  ,LUDLO  ,LMDLO  ,LTDLO  ,LHTDLO ,   NLDRV
     *VDLU   ,GDLU   ,PDLU   ,RDLU   ,ODLU   ,UDLU   ,MDLU   ,TDLU   ,HTDLU,     NLDRV
     *LVDLU  ,LGDLU  ,LPDLU  ,LRDLU  ,LODLU  ,LUDLU  ,LMDLU  ,LTDLU  ,LHTDLU,    NLDRV
     *VDLM   ,GDLM   ,PDLM   ,RDLM   ,ODLM   ,UDLM   ,MDLM   ,TDLM   ,HTDLM  ,   NLDRV
     *LVDLM  ,LGDLM  ,LPDLM  ,LRDLM  ,LODLM  ,LUDLM  ,LMDLM  ,LTDLM  ,LHTDLM ,   NLDRV
     *VDLT   ,GDLT   ,PDLT   ,RDLT   ,ODLT   ,UDLT   ,MDLT   ,TDLT   ,HTDLT  ,   NLDRV
     *LVDLT  ,LGDLT  ,LPDLT  ,LRDLT  ,LODLT  ,LUDLT  ,LMDLT  ,LTDLT  ,LHTDLT ,   NLDRV
     *VDLH   ,GDLH   ,PDLH   ,RDLH   ,ODLH   ,UDLH   ,MDLH   ,TDLH   ,HTDLH,     NLDRV
     *LVDLH  ,LGDLH  ,LPDLH  ,LRDLH  ,LODLH  ,LUDLH  ,LMDLH  ,LTDLH  ,LHTDLH      NLDRV
```

```
      DIMENSION JACOB(18, 18)                                           NLDRV
      EQUIVALENCE (JACOB, VDV)                                          NLDRV
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      COMMON /AXLE/                                                     AXLE
     *AV    ,AG    ,AP    ,AM    ,AVV   ,AGV   ,APV   ,AMV   ,AVG   ,   AXLE
     *AGG   ,APG   ,AMG   ,AVP   ,AGP   ,APP   ,AMP   ,AVR   ,AGR   ,   AXLE
     *APR   ,AMR   ,AVO   ,AGO   ,APO   ,AMO   ,AVU   ,AGU   ,APU   ,   AXLE
     *AMU   ,AVM   ,AGM   ,APM   ,AMM   ,AVZ   ,AGZ   ,APZ   ,AMZ   ,   AXLE
     *EAVV  ,EAGV  ,EAPV  ,EAMV  ,EAVG  ,EAGG  ,EAPG  ,EAMG  ,EAVP  ,   AXLE
     *EAGP  ,EAPP  ,EAMP  ,EAVR  ,EAGR  ,EAPR  ,EAMR  ,EAVO  ,EAGO  ,   AXLE
     *EAPO  ,EAMO  ,EAVU  ,EAGU  ,EAPU  ,EAMU  ,EAVM  ,EAGM  ,EAPM  ,   AXLE
     *EAMM  ,EAVZ  ,EAGZ  ,EAPZ  ,EAMZ  ,AVVV  ,AGVV  ,APVV  ,AMVV  ,   AXLE
     *AVGV  ,AGGV  ,APGV  ,AMGV  ,AVPV  ,AGPV  ,APPV  ,AMPV  ,AVRV  ,   AXLE
     *AGRV  ,APRV  ,AMRV  ,AVOV  ,AGOV  ,APOV  ,AMOV  ,AVUV  ,AGUV  ,   AXLE
     *APUV  ,AMUV  ,AVMV  ,AGMV  ,APMV  ,AMMV  ,AVZV  ,AGZV  ,APZV  ,   AXLE
     *AMZV  ,AVVG  ,AGVG  ,APVG  ,AMVG  ,AVGG  ,AGGG  ,APGG  ,AMGG  ,   AXLE
     *AVPG  ,AGPG  ,APPG  ,AMPG  ,AVRG  ,AGRG  ,APRG  ,AMRG  ,AVOG  ,   AXLE
     *AGOG  ,APOG  ,AMOG  ,AVUG  ,AGUG  ,APUG  ,AMUG  ,AVMG  ,AGMG  ,   AXLE
     *APMG  ,AMMG  ,AVZG  ,AGZG  ,APZG  ,AMZG  ,AVVP  ,AGVP  ,APVP  ,   AXLE
     *AMVP  ,AVGP  ,AGGP  ,APGP  ,AMGP  ,AVPP  ,AGPP  ,APPP  ,AMPP  ,   AXLE
     *AVRP  ,AGRP  ,APRP  ,AMRP  ,AVOP  ,AGOP  ,APOP  ,AMOP  ,AVUP  ,   AXLE
     *AGUP  ,APUP  ,AMUP  ,AVMP  ,AGMP  ,APMP  ,AMMP  ,AVZP  ,AGZP  ,   AXLE
     *APZP  ,AMZP  ,AVVR  ,AGVR  ,APVR  ,AMVR  ,AVGR  ,AGGR  ,APGR      AXLE
      COMMON /AXLE/                                                     AXLE
     *AMGR  ,AVPR  ,AGPR  ,APPR  ,AMPR  ,AVRR  ,AGRR  ,APRR  ,AMRR  ,   AXLE
     *AVOR  ,AGOR  ,APOR  ,AMOR  ,AVUR  ,AGUR  ,APUR  ,AMUR  ,AVMR  ,   AXLE
     *AGMR  ,APMR  ,AMMR  ,AVZR  ,AGZR  ,APZR  ,AMZR  ,AVVO  ,AGVO  ,   AXLE
```

```
     *APVO  ,AMVO  ,AVGO  ,AGGO  ,APGO  ,AMGO  ,AVPO  ,AGPO  ,APPO  ,     AXLE
     *AMPO  ,AVRO  ,AGRO  ,APRO  ,AMRO  ,AVOO  ,AGOO  ,APOO  ,AMOO  ,     AXLE
     *AVUO  ,AGUO  ,APUO  ,AMUO  ,AVMO  ,AGMO  ,APMO  ,AMMO  ,AVZO  ,     AXLE
     *AGZO  ,APZO  ,AMZO  ,AVVU  ,AGVU  ,APVU  ,AMVU  ,AVGU  ,AGGU  ,     AXLE
     *APGU  ,AMGU  ,AVPU  ,AGPU  ,APPU  ,AMPU  ,AVRU  ,AGRU  ,APRU  ,     AXLE
     *AMRU  ,AVOU  ,AGOU  ,APOU  ,AMOU  ,AVUU  ,AGUU  ,APUU  ,AMUU  ,     AXLE
     *AVMU  ,AGMU  ,APMU  ,AMMU  ,AVZU  ,AGZU  ,APZU  ,AMZU  ,AVVM  ,     AXLE
     *AGVM  ,APVM  ,AMVM  ,AVGM  ,AGGM  ,APGM  ,AMGM  ,AVPM  ,AGPM  ,     AXLE
     *APPM  ,AMPM  ,AVRM  ,AGRM  ,APRM  ,AMRM  ,AVOM  ,AGOM  ,APOM  ,     AXLE
     *AMOM  ,AVUM  ,AGUM  ,APUM  ,AMUM  ,AVMM  ,AGMM  ,APMM  ,AMMM  ,     AXLE
     *AVZM  ,AGZM  ,APZM  ,AMZM  ,AVVZ  ,AGVZ  ,APVZ  ,AMVZ  ,AVGZ  ,     AXLE
     *AGGZ  ,APGZ  ,AMGZ  ,AVPZ  ,AGPZ  ,APPZ  ,AMPZ  ,AVRZ  ,AGRZ  ,     AXLE
     *APRZ  ,AMRZ  ,AVOZ  ,AGOZ  ,APOZ  ,AMOZ  ,AVUZ  ,AGUZ  ,APUZ  ,     AXLE
     *AMUZ  ,AVMZ  ,AGMZ  ,APMZ  ,AMMZ  ,AVZZ  ,AGZZ  ,APZZ  ,AMZZ        AXLE
      COMMON /AXLE/                                                        AXLE
     *AVLV  ,AGLV  ,APLV  ,AMLV  ,AVLG  ,AGLG  ,APLG  ,AMLG  ,AVLP  ,     AXLE
     *AGLP  ,APLP  ,AMLP  ,AVVLV ,AGVLV ,APVLV ,AMVLV ,AVGLV ,AGGLV ,     AXLE
     *APGLV ,AMGLV ,AVPLV ,AGPLV ,APPLV ,AMPLV ,AVRLV ,AGRLV ,APRLV ,     AXLE
     *AMRLV ,AVOLV ,AGOLV ,APOLV ,AMOLV ,AVULV ,AGULV ,APULV ,AMULV ,     AXLE
     *AVMLV ,AGMLV ,APMLV ,AMMLV ,AVZLV ,AGZLV ,APZLV ,AMZLV ,AVVLG ,     AXLE
     *AGVLG ,APVLG ,AMVLG ,AVGLG ,AGGLG ,APGLG ,AMGLG ,AVPLG ,AGPLG ,     AXLE
     *APPLG ,AMPLG ,AVRLG ,AGRLG ,APRLG ,AMRLG ,AVOLG ,AGOLG ,APOLG ,     AXLE
     *AMOLG ,AVULG ,AGULG ,APULG ,AMULG ,AVMLG ,AGMLG ,APMLG ,AMMLG ,     AXLE
     *AVZLG ,AGZLG ,APZLG ,AMZLG ,AVVLP ,AGVLP ,APVLP ,AMVLP ,AVGLP ,     AXLE
     *AGGLP ,APGLP ,AMGLP ,AVPLP ,AGPLP ,APPLP ,AMPLP ,AVRLP ,AGRLP ,     AXLE
     *APRLP ,AMRLP ,AVOLP ,AGOLP ,APOLP ,AMOLP ,AVULP ,AGULP ,APULP ,     AXLE
     *AMULP ,AVMLP ,AGMLP ,APMLP ,AMMLP ,AVZLP ,AGZLP ,APZLP ,AMZLP        AXLE
      COMMON /PC/                                                          PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX         PC
C                                                                          NLDRV
      ASF1(A,B) = LMTAU*A + LVTAU*B                                        NLDRV
      ASF2(C,D) = LGAM*C  + LPSI*D                                         NLDRV
C                                                                          NLDRV
      DO 1 I = 1,N                                                         NLDRV
    1 NOM(I)=Z(I)                                                          NLDRV
      CALL FORCES                                                          NLDRV
      IF(KODE .EQ. KODES) GO TO 3                                          NLDRV  3
      KODES = KODE                                                         NLDRV
      DO 2 I = 1, 18                                                       NLDRV
      DO 2 J = 1, 18                                                       NLDRV
    2 JACOB(I, J) = 0.                                                     NLDRV
    3 IF(KODE .NE. 0) CALL APPLY                                           NLDRV
C                                                                          NLDRV
C                                                                          NLDRV
C          EVALUATE INTERMEDIATE CONSTANTS                                 NLDRV
C                                                                          NLDRV
      CGV = COSGAM * V                                                     NLDRV
      RCRHO = R * COSRHO                                                   NLDRV
      OM2RVC = OMEGA2 * R * COSRHO / V                                     NLDRV
      ORVCOG = OM2RVC/COSGAM                                               NLDRV
      OM2RCO = OMEGA2 * RCRHO                                              NLDRV
      OM2R = OMEGA2*R                                                      NLDRV
      OM2CO = OMEGA2*COSRHO                                                NLDRV
C                                                                          NLDRV
      V2 = V*V                                                             NLDRV
      R2 = R*R                                                             NLDRV
      GOV = G/V                                                            NLDRV
      VOR = V/R                                                            NLDRV
C                                                                          NLDRV
      TAUV = TAU / V                                                       NLDRV
      TAUVR = -TAU*V/R                                                     NLDRV
      TVRO = TAU*V/RCRHO                                                   NLDRV
      TAUCGV = TAU/CGV                                                     NLDRV
      LOTR = LRHO*TAU/R                                                    NLDRV
      LVTAU = -LV*TAU                                                      NLDRV
      LMTAU = -LM*TAU                                                      NLDRV
      LURC = LMU/RCRHO                                                     NLDRV
      LURCT = LURC*TAU                                                     NLDRV
      LUTVRC = LURC*TAU*V                                                  NLDRV
      LGTV = LGAM*TAU/V                                                    NLDRV
      LPTCGV = LPSI*TAU/CGV                                                NLDRV
      LPTCG  = LPSI*TAU/COSGAM                                             NLDRV
      LOVTR  = LRHO*V*TAU/R                                                NLDRV
C                                                                          NLDRV
      CGCP = COSGAM * COSPSI                                               NLDRV
```

```
      CGSP = COSGAM * SINPSI                                            NLDRV
      COCG = COSRHO * COSGAM                                            NLDRV
      SOSG = SINRHO * SINGAM                                            NLDRV
      COSG = COSRHO * SINGAM                                            NLDRV
      SPSO = SINPSI * SINRHO                                            NLDRV
      SPSOSG = SPSO * SINGAM                                            NLDRV
      CPCO  = COSPSI* COSRHO                                            NLDRV
      COSP = COSRHO*SINPSI                                              NLDRV
      CPSO = COSPSI*SINRHO                                              NLDRV
      SPSG = SINPSI*SINGAM                                              NLDRV
      CPSG = COSPSI*SINGAM                                              NLDRV
      SGOCG  = SINGAM/COSGAM                                            NLDRV
      SOOCO = SINRHO/COSRHO                                             NLDRV
      CG2 = COSGAM*COSGAM                                               NLDRV
      CO2 = COSRHO*COSRHO                                               NLDRV
C                                                                       NLDRV
      EXPRES = SINRHO*SINGAM*COSPSI + COSRHO*COSGAM                     NLDRV
      EXPRS1 = COSRHO*SINGAM - CGCP*SINRHO                              NLDRV
      EXPRS2 = CPSG*COS2RO - SIN2RO*COSGAM                              NLDRV
      EXPRS3 = COS2RO*CGCP + SIN2RO*SINGAM                              NLDRV
C                     ASSUME FREE FALL.  COMP  1ST PARTS. W/RESP.TO STATE NLDRV
C                     OF ALL STATE EQS. EXCEPT GD AND PD.               NLDRV
      HTDT = QMULT*17600.*SQRT(RO/RHOB)*(V/26000.)**3.15                NLDRV
      RDV = TAU * SINGAM                                                NLDRV
      ODV =(TAU/R) * CGCP                                               NLDRV
      UDV = (TAU/RCRHO) * CGSP                                          NLDRV
      HTDV = TAU*HTDT*3.15/V                                            NLDRV
      VDG = TAU*(OM2RCO*EXPRES - COSGAM*G)                              NLDRV
      RDG = TAU*CGV                                                     NLDRV
      ODG = TAUVR*CPSG                                                  NLDRV
      UDG = -TVRO*SPSG                                                  NLDRV
      VDP = TAU*OM2RCO*CGSP*SINRHO                                      NLDRV
      ODP =      TAUVR*CGSP                                             NLDRV
      UDP =  TVRO * CGCP                                                NLDRV
      VDR = TAU*(OM2CO*EXPRS1 - GH*SINGAM)                              NLDRV
      ODR =  TAUVR * CGCP/R                                             NLDRV
      UDR =  (-TVRO/R)     * CGSP                                       NLDRV
      IF(IATM .LT. 2) HTDR = TAU*HTDT*ROR/RO/2.                         NLDRV
      VDO = -TAU*OM2R*EXPRS3                                            NLDRV
      UDO = TVRO*CGSP*SOOCO                                             NLDRV
      VDT = OM2RCO*EXPRS1 - G*SINGAM                                    NLDRV
      RDT = SINGAM * V                                                  NLDRV
      ODT = CGCP * VOR                                                  NLDRV
      UDT = CGSP * V / RCRHO                                            NLDRV
C                     IS THIS VERT. RISE/PITCHOVER                      NLDRV
      IF(KODE .NE. 4) GO TO 1000                                        NLDRV   1000
C                     YES.  COMP. 1ST PART. OF GD W/RESP. TO TAU        NLDRV
      GDT = GAMMAD                                                      NLDRV
C                     COMPUTE STATE DERIVATIVES                         NLDRV
      VD = TAU * (AV + VDT)                                             NLDRV
      GD = TAU * GDT                                                    NLDRV
      PD = 0.                                                           NLDRV
      RD = TAU*RDT                                                      NLDRV
      OD = TAU*ODT                                                      NLDRV
      UD = TAU*UDT                                                      NLDRV
      MD = TAU*AM                                                       NLDRV
      HTD = TAU*HTDT                                                    NLDRV
C                     ACCOUNT FOR APPLIED LOAD EFFECTS IN 1ST PARTS.    NLDRV
C                     W/RESP. TO STATE OF STATE EQS.                    NLDRV
      VDV = AVV  * TAU                                                  NLDRV
      MDV = AMV  * TAU                                                  NLDRV
      MDG = AMG  * TAU                                                  NLDRV
      MDP = AMP  * TAU                                                  NLDRV
      MDR = AMR  * TAU                                                  NLDRV
      MDO = AMO  * TAU                                                  NLDRV
      VDM = AVM  * TAU                                                  NLDRV
      MDM = AMM  * TAU                                                  NLDRV
      MDT = AM + AMZ*TAU                                                NLDRV
      VDG = VDG + AVG * TAU                                             NLDRV
      VDP = VDP + AVP * TAU                                             NLDRV
      VDR = VDR + AVR * TAU                                             NLDRV
      VDO = VDO + AVO * TAU                                             NLDRV
      VDT = VDT + AV + AVZ*TAU                                          NLDRV
C                     COMPUTE COSTATE DERIVATIVES.                      NLDRV
      LVD =   -LV*VDV - LR*RDV - LRHO*ODV - LMU*UDV - LM*MDV - LHT*HTDV NLDRV
```

```
      LGD =    -LV*VDG  - LR*RDG - LRHO*ODG - LMU*UDG - LM*MDG                NLDRV
      LPD =    -LV*VDP  - LRHO*ODP -LMU*UDP - LM*MDP                          NLDRV
      LRD =    -LV*VDR - LRHO*ODR - LMU*UDR - LM*MDR - LHT*HTDR               NLDRV
      LOD =    -LV*VDO  -LMU*UDO - LM*MDO                                     NLDRV
      LMD =    -LV*VDM - LM*MDM                                               NLDRV
      LTD =    -LV*VDT - LGAM*GDT -LR*RDT - LRHO*ODT - LMU*UDT -LM*MDT        NLDRV
     *         -LHT*HTDT                                                      NLDRV
C                                                                             NLDRV
      GO TO 9000                                                              NLDRV  9000
C                    NOT VERT. RISE/PITCHOVER.   STILL ASSUMING FREE          NLDRV
C                    FALL, COMP. 1ST PARTS. W/RESP. TO STATE OF GD AND        NLDRV
C                    PD.                                                      NLDRV
 1000 GDV =TAUV * (COSGAM*(GOV +VOR) - OM2RVC*EXPRES)                         NLDRV
      PDV =(TAU*SPSO) * (COSGAM/RCRHO - ORVCOG/V  )                           NLDRV
      GDG = TAU*(-OM2RVC*EXPRS1- SINGAM*(VOR - GOV))                          NLDRV
      PDG = TAU * (SPSOSG*(ORVCOG/COSGAM - V/RCRHO) - CPCO*OMEGAT/CG2)        NLDRV
      GDP = TAU * (      CPCO * OMEGAT- OM2RVC * SPSOSG)                      NLDRV
      PDP = TAU*(OMEGAT*COSP*SGOCG + CPSO*ORVCOG + CGCP*SOOCO*VOR)            NLDRV
      GDR = TAU * ( EXPRES*OM2CO/V - COSGAM*(VOR/R + GH/V))                   NLDRV
      PDR = TAU*SPSO/R * (ORVCOG-  CGV/RCRHO)                                 NLDRV
      GDO = TAU*(EXPRS2*OM2R/V - OMEGAT*SPSO)                                 NLDRV
      PDO = TAU*(OMEGAT*(CPSO*SGOCG + COSRHO) + SINPSI*OM2R*COS2RO/CGV        NLDRV
     *          +    CGSP*VOR/CO2)                                            NLDRV
      GDT = OMEGAT*COSP + EXPRES*OM2RVC + COSGAM*(VOR - GOV)                  NLDRV
      PDT = OMEGAT*(SINRHO - CPCO*SGOCG) + SPSO*(ORVCOG + CGV/RCRHO)          NLDRV
C                    IS THIS FREE FALL                                        NLDRV
      IF(KODE.NE.0) GO TO 2000                                                NLDRV  2000
C                    COMPUTE STATE AND COSTATE DERIVATIVES.                   NLDRV
      VD = TAU * VDT                                                          NLDRV
      GD = TAU * GDT                                                          NLDRV
      PD = TAU * PDT                                                          NLDRV
      RD = TAU * RDT                                                          NLDRV
      OD = TAU * OUT                                                          NLDRV
      UD = TAU * UDT                                                          NLDRV
      MD = 0.                                                                 NLDRV
      HTD = 0.                                                                NLDRV
      LVD = -LGAM * GDV - LPSI*PDV - LR*RDV - LRHO*ODV - LMU*UDV              NLDRV
      LGD = -LV*VDG - LGAM*GDG - LPSI*PDG - LR*RDG - LRHO*ODG - LMU*UDG       NLDRV
      LPD = -LV*VDP - LGAM*GDP - LPSI*PDP - LMU*UDP - LRHO*ODP                NLDRV
      LRD = -LV*VDR - LGAM*GDR - LPSI*PDR - LRHO*ODR - LMU*UDR                NLDRV
      LOD = -LV*VDO - LGAM*GDO - LPSI*PDO - LMU*UDO                           NLDRV
      LMD = 0.                                                                NLDRV
      LTD = -LV*VDT - LGAM*GDT - LPSI*PDT - LR*RDT - LRHO*ODT - LMU*UDT       NLDRV
C                                                                             NLDRV
      GO TO 9000                                                              NLDRV  9000
C                    NOT FREE FALL.  COMPUTE STATE DERIVATIVES.               NLDRV
 2000 VD = TAU*(AV + VDT)                                                     NLDRV
      GD = TAU*(AG/V + GDT)                                                   NLDRV
      PD = TAU*(AP/   CGV     + PDT)                                          NLDRV
      RD = TAU*RDT                                                            NLDRV
      OD = TAU*ODT                                                            NLDRV
      UD = TAU*UDT                                                            NLDRV
      MD = TAU*AM                                                             NLDRV
      HTD = TAU*HTDT                                                          NLDRV
C                    IS THIS NON-OPTIMAL CONTROL                              NLDRV
      IF(KODE .LT. 4) GO TO 3000                                              NLDRV  3000
C                    YES.  ACCOUNT FOR APPLIED LOAD EFFECTS ON 1ST PARTS.     NLDRV
C                    W/RESP. TO STATE OF STATE EQS.                           NLDRV
      VDV =  AVV * TAU                                                        NLDRV
      VDM =  AVM * TAU                                                        NLDRV
      GDM =  AGM * TAUV                                                       NLDRV
      PDM =  APM * TAUCGV                                                     NLDRV
      MDV =  AMV * TAU                                                        NLDRV
      MDR =  AMR * TAU                                                        NLDRV
      MDM =  AMM * TAU                                                        NLDRV
      MDT = AM + AMZ*TAU                                                      NLDRV
C                                                                             NLDRV
      GDV = GDV + TAUV*(AGV-AG/V)                                             NLDRV
      PDV  = PDV + (APV-AP/V)* TAUCGV                                         NLDRV
      PDG = PDG + TAUCGV*SGOCG*AP                                             NLDRV
      VDR  = VDR + AVR*TAU                                                    NLDRV
      GDR  = GDR + AGR*TAUV                                                   NLDRV
      PDR  = PDR + APR*TAUCGV                                                 NLDRV
      VDT = VDT + AV + AVZ*TAU                                                NLDRV
```

```
      GDT = GDT + (AGZ*TAU + AG)/V                                      NLDRV
      PDT = PDT + (APZ*TAU + AP)/CGV                                    NLDRV
C                    COMPUTE COSTATE EQS. FOR NON-OPTIMAL CONTROL.      NLDRV
      LVD = -LV*VDV - LGAM*GDV - LPSI*PDV - LR*RDV - LRHO*ODV - LMU*UDV NLDRV
     *        -LM*MDV - LHT*HTDV                                        NLDRV
      LGD = -LV*VDG - LGAM*GDG - LPSI*PDG - LR*RDG - LRHO*ODG - LMU*UDG NLDRV
      LPD = -LV*VDP - LGAM*GDP - LPSI*PDP          - LRHO*ODP - LMU*UDP NLDRV
      LRD = -LV*VDR - LGAM*GDR - LPSI*PDR - LM*MDR - LRHO*ODR - LMU*UDR NLDRV
     *        -LHT*HTDR                                                 NLDRV
      LOD = -LV*VDO - LGAM*GDO - LPSI*PDO                     - LMU*UDO NLDRV
      LMD = -LV*VDM - LGAM*GDM - LPSI*PDM - LM*MDM                      NLDRV
      LTD = -LV*VDT - LGAM*GDT - LPSI*PDT - LM*MDT            - LMU*UDT NLDRV
     *        -LR*RDT - LRHO*ODT - LHT*HTDT                             NLDRV
C                                                                       NLDRV
C                                                                       NLDRV
      GO TO 9000                                                        NLDRV- 9000
C                   COMPUTE COSTATE EQS. FOR OPTIMAL CONTROL            NLDRV
 3000 LVD = -LV*TAU*EAVV - LGAM*(GDV+TAUV*(EAGV-AG/V))                  NLDRV
     *        -LPSI*(PDV+TAUCGV    *(EAPV-AP/V)  ) - LR*RDV - LRHO*ODV  NLDRV
     *        -LMU*UDV - LM*EAMV*TAU - LHT*HTDV                         NLDRV
      LGD = -LV*VDG - LGAM*GDG - LPSI*(PDG + TAUCGV*AP*SGOCG)           NLDRV
     *        -LR*RDG - LRHO*ODG - LMU*UDG                              NLDRV
      LPD = -LV*VDP - LGAM*GDP - LPSI*PDP - LRHO*ODP - LMU*UDP          NLDRV
      LRD = -LV*(VDR + TAU*EAVR)- LGAM*(GDR + TAUV*EAGR)                NLDRV
     *        -LPSI*(PDR+TAUCGV*EAPR  ) - LRHO*ODR - LMU*UDR -LM*TAU*EAMR NLDRV
     *        -LHT*HTDR                                                 NLDRV
      LOD = -LV*VDO - LGAM*GDO - LPSI*PDO - LMU*UDO                     NLDRV
      LMD =  TAU*(-LV*EAVM -(LGAM*EAGM+LPSI*EAPM/COSGAM)/V - LM*EAMM)   NLDRV
      LTD = -LV*(VDT + AV + TAU*EAVZ) - LGAM*(GDT +(AG + TAU*EAGZ)/V)   NLDRV
     *        -LPSI*(PDT +(AP + TAU*EAPZ)/CGV) - LR*RDT - LRHO*ODT-LMU*UDT NLDRV
     *        -LM*(AM + TAU*EAMZ) - LHT*HTDT                            NLDRV
C                   IS QL CONVERGED                                     NLDRV
      IF(KONVER) GO TO 9000                                             NLDRV 9000
C                   NO.  OPTIMAL CONTROL IN EFFECT.  COMPUTE 1ST PARTS. NLDRV
C                   W/RESP. TO COSTATE OF COSTATE EQS.                  NLDRV
      LPDLV = -VDP                                                      NLDRV
      LODLV = -VDO                                                      NLDRV
      LPDLG = -GDP                                                      NLDRV
      LODLG = -GDO                                                      NLDRV
      LPDLP = -PDP                                                      NLDRV
      LODLP = -PDO                                                      NLDRV
      LVDLR = -RDV                                                      NLDRV
      LGDLR = -RDG                                                      NLDRV
      LTDLR = -RDT                                                      NLDRV
      LVDLO = -ODV                                                      NLDRV
      LGDLO = -ODG                                                      NLDRV
      LPDLO = -ODP                                                      NLDRV
      LRDLO = -ODR                                                      NLDRV
      LTDLO = -ODT                                                      NLDRV
      LVDLU = -UDV                                                      NLDRV
      LGDLU = -UDG                                                      NLDRV
      LPDLU = -UDP                                                      NLDRV
      LRDLU = -UDR                                                      NLDRV
      LODLU = -UDO                                                      NLDRV
      LTDLU = -UDT                                                      NLDRV
      LVDLV =  ASF1( AMVLV, AVVLV) - EAVV*TAU + LGTV*(AGLV/V -  AGVLV) + NLDRV
     *         LPTCGV*(APLV/V -  APVLV)                                 NLDRV
C                                                                       NLDRV
      LGDLV = -LPTCGV*APLV*SGOCG   - VDG                                NLDRV
C                                                                       NLDRV
      LRDLV =  ASF1( AMRLV, AVRLV) - EAVR*TAU - LGTV* AGRLV             NLDRV
     *                             - LPTCGV* APRLV - VDR                NLDRV
C                                                                       NLDRV
      LMDLV =  ASF1( AMMLV, AVMLV) - EAVM*TAU - LGTV* AGMLV             NLDRV
     *                             -LPTCGV* APMLV                       NLDRV
C                                                                       NLDRV
      LTDLV =  ASF1( AMZLV, AVZLV) - EAVZ*TAU- LGTV* AGZLV              NLDRV
     *                             - LPTCGV* APZLV - VDT                NLDRV
     *        - AV  - LV*AVLV  -  LM*AMLV - LGAM*AGLV/V - LPSI*APLV/CGV NLDRV
C                                                                       NLDRV
      LVDLG =  ASF1( AMVLG, AVVLG)  +  LGTV*(AGLG/V -  AGVLG)           NLDRV
     *         + LPTCGV*(APLG/V- APVLG) + TAUV*(AG/V - EAGV) - GDV      NLDRV
C                                                                       NLDRV
      LGDLG  =  -LPTCGV*APLG*SGOCG - GDG                                NLDRV
C                                                                       NLDRV
```

```
      LRDLG  =  ASF1( AMRLG, AVRLG)  -  LGTV* AGRLG  - LPTCGV* APRLG      NLDRV
     *          - TAUV*EAGR  - GDR                                        NLDRV
C                                                                         NLDRV
      LMDLG  =  ASF1( AMMLG, AVMLG)  -  LGTV* AGMLG - LPTCGV* APMLG       NLDRV
     *          - TAUV*EAGM                                               NLDRV
C                                                                         NLDRV
      LTDLG  =  ASF1( AMZLG, AVZLG)  -  LGTV* AGZLG -  LPTCGV* APZLG      NLDRV
     *          - TAUV*EAGZ - LV*AVLG- LM*AMLG- (LGAM*AGLG + AG)/V        NLDRV
     *          - LPSI*APLG/CGV - GDT                                     NLDRV
C                                                                         NLDRV
      LVDLP = ASF1( AMVLP, AVVLP)  +  LGTV*(AGLP/V -  AGVLP)              NLDRV
     *        + LPTCGV*(APLP/V -  APVLP) +  TAUCGV*(AP/V - EAPV) - PDV    NLDRV
C                                                                         NLDRV
      LGDLP =  - LPTCGV*APLP*SGOCG - TAUCGV*AP*SGOCG       - PDG          NLDRV
C                                                                         NLDRV
      LRDLP =  ASF1( AMRLP, AVRLP) - LGTV* AGRLP - LPTCGV* APRLP  - PDR   NLDRV
     *        - TAUCGV*EAPR                                               NLDRV
C                                                                         NLDRV
      LMDLP =  ASF1( AMMLP, AVMLP) - LGTV* AGMLP - LPTCGV* APMLP          NLDRV
     *        - TAUCGV*EAPM                                               NLDRV
C                                                                         NLDRV
      LTDLP =  ASF1( AMZLP, AVZLP) - LGTV* AGZLP - LPTCGV* APZLP          NLDRV
     *        - TAUCGV*EAPZ - LV*AVLP  - LM*AMLP - LGAM*AGLP/V            NLDRV
     *        - LPSI*APLP/CGV     -AP/CGV - PDT                           NLDRV
C                                                                         NLDRV
      LVDLM  =   -TAU*EAMV                                                NLDRV
C                                                                         NLDRV
      LRDLM  =   -TAU*EAMR                                                NLDRV
C                                                                         NLDRV
      LMDLM  =   -TAU*EAMM                                                NLDRV
C                                                                         NLDRV
      LTDLM  =   -TAU*EAMZ - AM                                           NLDRV
      LVDLH = -HTDV                                                       NLDRV
      LRDLH = -HTDR                                                       NLDRV
      LTDLH = -HTDT                                                       NLDRV
C                   ACCOUNT FOR THE EFFECTS OF APPLIED LOADS ON THE       NLDRV
C                   TOTAL 1ST PARTS. W/RESP. TO STATE OF STATE EQS.       NLDRV
 3100 GDV = GDV +(TAUV )*(AGV - AG/V)                                     NLDRV
      PDV = PDV + (TAUCGV )* (APV - AP/V)                                 NLDRV
      VDG = VDG + AVG *TAU                                                NLDRV
      GDG = GDG + AGG *TAUV                                               NLDRV
      PDG = PDG + TAUCGV*(AP*SGOCG + APG)                                 NLDRV
      VDR = VDR + AVR *TAU                                                NLDRV
      GDR = GDR + AGR * TAUV                                              NLDRV
      PDR = PDR + APR*TAUCGV                                              NLDRV
      VDT = VDT + AV + AVZ*TAU                                            NLDRV
      GDT = GDT + (AGZ*TAU + AG)/V                                        NLDRV
      PDT = PDT + (APZ*TAU + AP)/CGV                                      NLDRV
      VDV = AVV*TAU                                                       NLDRV
      MDV = AMV*TAU                                                       NLDRV
      MDG = AMG*TAU                                                       NLDRV
      MDR = AMR*TAU                                                       NLDRV
      VDM = AVM*TAU                                                       NLDRV
      GDM = AGM*TAUV                                                      NLDRV
      PDM = APM*TAUCGV                                                    NLDRV
      MDM = AMM*TAU                                                       NLDRV
      MDT = AM + AMZ*TAU                                                  NLDRV
C                  MOVE STATE AND COSTATE DERIVATIVES INTO OUTPUT         NLDRV
C                  ARRAY.                                                 NLDRV
 9000 CONTINUE                                                            NLDRV
      ZD( 1) =  VD                                                        NLDRV
      ZD( 2) =  GD                                                        NLDRV
      ZD( 3) =  PD                                                        NLDRV
      ZD( 4) =  RD                                                        NLDRV
      ZD( 5) =  OD                                                        NLDRV
      ZD( 6) =  UD                                                        NLDRV
      ZD( 7) =  MD                                                        NLDRV
      ZD( 8) =  0.                                                        NLDRV
      ZD(9) = HTD                                                         NLDRV
      ZD(10)= LVD                                                         NLDRV
      ZD(11)= LGD                                                         NLDRV
      ZD(12)= LPD                                                         NLDRV
      ZD(13)= LRD                                                         NLDRV
      ZD(14)= LOD                                                         NLDRV
      ZD(15)= 0.                                                          NLDRV
```

```
      ZD(16)= LMD                                                    NLDRV
      ZD(17)= LTD                                                    NLDRV
      ZD(18)= 0.                                                     NLDRV
C                  IS QL CONVERGED.                                  NLDRV
      IF(.NOT KONVER) GO TO 9001                                     NLDRV   9001
C                  COMPUTE AND STORE VELOCITY LOSS DERIVATIVES.      NLDRV
      ZD(19) = FVAC/M*TAU                                            NLDRV
      ZD(20) = (DRAG + DB*COSA)/M*TAU                                NLDRV
      ZD(21) = AE*PA*CODAE/M*TAU                                     NLDRV
      ZD(22) = G*SINGAM*TAU                                          NLDRV
      ZD(23) = FVAC*(1. - CODAE)/M*TAU                               NLDRV
      RETURN                                                         NLDRV
C                  QL NOT CONVERGED.  ASSUME FREE FALL AND COMPUTE 1ST NLDRV
C                  PARTS.W/RESP. TO STATE OF COSTATE EQS. OMITTING   NLDRV
C                  TERMS WITH LGAM OR LPSI.                          NLDRV
 9001 LVDG  =  LOTR*CPSG - LR*TAU*COSGAM + LURCT*SPSG               NLDRV
      LVDP  =  LOTR*CGSP  -  LURCT*CGCP                              NLDRV
      LVDR = (LOTR*CGCP + LURCT*CGSP)/R - LHT*3.15/V*HTDR            NLDRV
      LVDO  = -LURCT*CGSP*SOOCO                                      NLDRV
      LVDT  = -LRHO*CGCP/R -     LR*SINGAM - LURC*CGSP               NLDRV
      LGDG  = LUTVRC*CGSP + LVTAU*(SINGAM*G - OM2RCO*EXPRS1)         NLDRV
     *                  + LR*TAU*SINGAM*V  + LOVTR*CGCP              NLDRV
C                                                                    NLDRV
      LGDP  = LUTVRC*CPSG  - LVTAU*OM2RCO*SPSOSG - LOVTR*SPSG        NLDRV
C                                                                    NLDRV
      LGDR  =-LUTVRC*SPSG/R + LVTAU*(EXPRES*OM2CO - GH*COSGAM) -     NLDRV
     *                               LOVTR*CPSG/R                    NLDRV
C                                                                    NLDRV
      LGDO = LUTVRC*SPSOSG/COSRHO + LVTAU*OM2R*EXPRS2                NLDRV
C                                                                    NLDRV
      LGDT  =LURC*V*SPSG        -LV*(OM2RCO*EXPRES - COSGAM*G)       NLDRV
     *                     - LR*CGV   + LRHO*VOR*CPSG                NLDRV
C                                                                    NLDRV
      LPDP  = LUTVRC*CGSP+(LVTAU*OM2RCO      *SINRHO +LOVTR)*CGCP    NLDRV
C                                                                    NLDRV
      LPDR = LVTAU*OM2CO*COSGAM*SPSO + (LUTVRC*CGCP - LOVTR*CGSP)/R  NLDRV
C                                                                    NLDRV
      LPDO  = LVTAU*OM2R*COS2RO*CGSP - LUTVRC*CGCP*SOOCO             NLDRV
C                                                                    NLDRV
      LPDT  = CGSP*(LRHO*VOR - LV*OM2RCO*SINRHO) - LURC*V*CGCP       NLDRV
C                                                                    NLDRV
      LRDR =-2./R2*(LUTVRC*CGSP + LOVTR*CGCP) - LVTAU*GRR*SINGAM     NLDRV
      IF(IATM .LT. 2) LRDR = LRDR - LHT*.5/(RO*ROR)*(2.*RO*RORR-ROR**2)  NLDRV
C                                                                    NLDRV
      LRDO  = LUTVRC*CGSP*SINRHO/RCRHO - LVTAU*OMEGA2*EXPRS3         NLDRV
      LRDT = LURC*VOR*CGSP - LV*(OM2CO*EXPRS1 - GH*SINGAM)           NLDRV
     *                        + LRHO*VOR*CGCP/R                      NLDRV
C                                                                    NLDRV
      LODO  = - LUTVRC*CGSP*SOOCO*SOOCO - LVTAU*OM2R*2.*(COS2RO*SINGAM - NLDRV
     *                                    SIN2RO*CGCP)               NLDRV
C                                                                    NLDRV
      LODT =-LURC*V*CGSP*SOOCO + LV*OM2R*EXPRS3                      NLDRV
C                  IS THIS VERT.RISE/PITCHOVER.                      NLDRV
      IF(KODE .NE. 4) GO TO 4000                                     NLDRV   4000
C                  YES.  ACCOUNT FOR EFFECTS OF APPLIED LOADS ON 1ST NLDRV
C                  PARTS.W/RESP.TO STATE OF COSTATE EQS.             NLDRV
      LVDV =     ASF1(AMVV,AVVV) - LHT*2.15/V*HTDV                   NLDRV
      LVDM =     ASF1(AMVM,AVVM)                                     NLDRV
      LGDM =     ASF1(AMGM,AVGM)                                     NLDRV
      LPDM =     ASF1(AMPM,AVPM)                                     NLDRV
      LRDM =     ASF1(AMRM,AVRM)                                     NLDRV
      LODM =     ASF1(AMOM,AVOM)                                     NLDRV
      LVDG = LVDG + ASF1(AMVG,AVVG)                                  NLDRV
      LVDP = LVDP + ASF1(AMVP,AVVP)                                  NLDRV
      LVDR = LVDR + ASF1(AMVR,AVVR)                                  NLDRV
      LVDO = LVDO + ASF1(AMVO,AVVO)                                  NLDRV
      LVDT = LVDT + ASF1(AMVZ,AVVZ) - LM*AMV - LV*AVV                NLDRV
      LGDG = LGDG + ASF1(AMGG,AVGG)                                  NLDRV
      LGDP = LGDP + ASF1(AMGP,AVGP)                                  NLDRV
      LGDR = LGDR + ASF1(AMGR,AVGR)                                  NLDRV
      LGDO = LGDO + ASF1(AMGO,AVGO)                                  NLDRV
      LGDT = LGDT + ASF1(AMGZ,AVGZ) - LM*AMG - LV*AVG                NLDRV
      LPDP = LPDP + ASF1(AMPP,AVPP)                                  NLDRV
      LPDR = LPDR + ASF1(AMPR,AVPR)                                  NLDRV
      LPDO = LPDO + ASF1(AMPO,AVPO)                                  NLDRV
```

```
      LPDT = LPDT + ASF1(AMPZ,AVPZ) - LM*AMP - LV*AVP                   NLDRV
C                                                                       NLDRV
      LRDR = LRDR + ASF1(AMRR,AVRR)                                     NLDRV
      LRDO = LRDO + ASF1(AMRO,AVRO)                                     NLDRV
      LRDT = LRDT + ASF1(AMRZ,AVRZ)  - LM*AMR - LV*AVR                  NLDRV
C                                                                       NLDRV
      LODO = LODO + ASF1(AMOO,AVOO)                                     NLDRV
      LODT = LODT + ASF1(AMOZ,AVOZ)  - LM*AMO - LV*AVO                  NLDRV
      LMDM = ASF1(AMMM, AVMM)                                           JUL19B
      LMDT = ASF1(AMMZ, AVMZ) - LM*AMM - LV*AVM                         JUL19B
      LTDT = ASF1(AMZZ, AVZZ) - 2.*(LM*AMZ + LV*AVZ)                    JUL19B
C                                                                       NLDRV
C                                                                       NLDRV
      GO TO 5000                                                        NLDRV   5000
C                   NOT VERT.RISE/PITCHOVER.  STILL ASSUMING FREE FALL, NLDRV
C                   ADD IN TERMS WITH LGAM AND LPSI TO THE 1ST PARTS.   NLDRV
C                   W/RESP. TO STATE OF THE COSTATE EQS.                NLDRV
 4000 LVDV = -TAU*2./V2 * ASF2(OM2RVC*EXPRES - COSGAM*GOV,              NLDRV
     *                        (ORVCOG*SPSO))                            NLDRV
     *           -LHT*2.15/V*HTDV                                       NLDRV
      LVDG = LVDG + TAU*ASF2((SINGAM*(GOV/V + 1./R) - OM2RVC*EXPRS1/V), NLDRV
     *                       (SPSOSG*(1./RCRHO + ORVCOG/CGV)))          NLDRV
C                                                                       NLDRV
      LVDP = LVDP + TAU*ASF2((-OM2RVC*SPSOSG/V),(CPSO*ORVCOG/V          NLDRV
     *                         - CGCP*SINRHO/RCRHO))                    NLDRV
C                                                                       NLDRV
      LVDR = LVDR + TAU*ASF2(EXPRES*OM2CO/V2 - COSGAM*(GH/V2 - 1./R2),  NLDRV
     *                       (SPSO*OM2CO/(CGV*V) +  CGSP*SOOCO/R2))     NLDRV
C                                                                       NLDRV
      LVDO = LVDO + TAU*ASF2(EXPRS2*OM2R/V2,                            NLDRV
     *                      (COS2RO*SINPSI*OM2R/(CGV*V) - CGSP/(R*CO2))) NLDRV
C                                                                       NLDRV
      LVDT = LVDT +     ASF2((EXPRES*OM2RVC/V - COSGAM*(GOV/V + 1./R)), NLDRV
     *                       (SPSO*(ORVCOG/V - COSGAM/RCRHO)))          NLDRV
C                                                                       NLDRV
C                                                                       NLDRV
      LGDG  =  LGDG + TAU*ASF2((EXPRES*OM2RVC + COSGAM*(VOR - GOV)),    NLDRV
     *                       (2.*OMEGAT*CPCO*SGOCG/CG2 -                NLDRV
     *                       SPSO*(OM2RVC*(2./CG2 - 1.)/COSGAM -        NLDRV
     *                       CGV/RCRHO)))                               NLDRV
C                                                                       NLDRV
      LGDP  =  LGDP + TAU*ASF2((OM2RVC*CGSP*SINRHO),  (CPSO*SINGAM *    NLDRV
     *                       (V/RCRHO-ORVCOG/COSGAM) - OMEGAT*COSP/CG2)) NLDRV
C                                                                       NLDRV
      LGDR  =  LGDR + TAU*ASF2((EXPRS1*OM2CO/V - SINGAM*(VOR/R + GH/V)), NLDRV
     *                       (-SPSOSG*(VOR/RCRHO + OM2CO/(CG2*V))))     NLDRV
C                                                                       NLDRV
      LGDO  =  LGDO + TAU*ASF2(( - OM2R*EXPRS3/V),(SPSG*VOR/CO2 -       NLDRV
     *                       (OMEGAT*CPSO + SPSG*OM2R*COS2RO/V)/CG2))   NLDRV
C                                                                       NLDRV
      LGDT  =  LGDT +     ASF2((OM2RVC*EXPRS1 + SINGAM*(VOR - GOV)),    NLDRV
     *                       ((OMEGAT*CPCO - SPSOSG*OM2RVC)/            NLDRV
     *                                  CG2 + SPSOSG*V/RCRHO))          NLDRV
C                                                                       NLDRV
C                                                                       NLDRV
      LPDP = LPDP + TAU*ASF2((OMEGAT*COSP+OM2RVC*CPSO*SINGAM),          NLDRV
     *                          (ORVCOG*SPSO - OMEGAT*CPCO*SGOCG +      NLDRV
     *                       CGV *SPSO/RCRHO))                          NLDRV
C                                                                       NLDRV
      LPDR = LPDR + TAU*ASF2((OM2CO*SPSOSG/V),(CPSO*(COSGAM*VOR/RCRHO - NLDRV
     *                       OM2CO/CGV)))                               NLDRV
C                                                                       NLDRV
      LPDO = LPDO + TAU*ASF2((OMEGAT*CPSO + SPSG*COS2RO*OM2R/V),        NLDRV
     *                       (OMEGAT*SPSOSG/COSGAM - COSPSI*OM2R*       NLDRV
     *                        COS2RO/CGV - CGCP*VOR/CO2))               NLDRV
C                                                                       NLDRV
      LPDT = LPDT +     ASF2((OM2RVC*SPSOSG - OMEGAT*CPCO),             NLDRV
     *                       -(OMEGAT*COSP*SGOCG + CPSO*ORVCOG +        NLDRV
     *                       CGCP*SOOCO*VOR))                           NLDRV
C                                                                       NLDRV
C                                                                       NLDRV
      LRDR = LRDR + TAU*ASF2((COSGAM*(GRR/V-2.*VOR/R2)),                NLDRV
     *                        (-2.*CGV*SPSO/(RCRHO*R2)))                NLDRV
C                                                                       NLDRV
      LRDO = LRDO + TAU*ASF2(-EXPRS2*OMEGA2/V ,(CGSP*V/(RCRHO*RCRHO) -  NLDRV
```

```
     *                                  SINPSI*COS2RO*OMEGA2/CGV))          NLDRV
C                                                                           NLDRV
      LRDT = LRDT +      ASF2((COSGAM*(VOR/R + GH/V) - EXPRES*OM2CO/V) ,    NLDRV
     *                              (SPSO*(CGV/(RCRHO*R) - OM2CO/CGV)))     NLDRV
C                                                                           NLDRV
C                                                                           NLDRV
      LODO  = LODO  + TAU*ASF2((OMEGAT*COSP + 2.*OM2R/V *                   NLDRV
     *                          (COS2RO*COSGAM + SIN2RO*CPSG)),             NLDRV
     *                          (OMEGAT*(SINRHO-CPCO*SGOCG) +               NLDRV
     *                         2.*SPSO*(2.*ORVCOG - CGV/(RCRHO*CO2))))      NLDRV
C                                                                           NLDRV
      LODT  = LODT  +      ASF2((OMEGAT*SPSO - EXPRS2*OM2R/V),              NLDRV
     *                         - (OMEGAT*(CPSO*SGOCG + COSRHO) +            NLDRV
     *                         SINPSI*COS2RO*OM2R/CGV + CGSP*VOR/CO2))      NLDRV
C                     IS THIS FREE FALL                                     NLDRV
      IF(KODE.EQ.0) GO TO 5000                                              NLDRV
C                     NOT FREE FALL.  IS THIS OPTIMAL CONTROL.              NLDRV
      IF(KODE .LT. 4) GO TO 4500                                            NLDRV
C                     NON-OPTIMAL.  ACCOUNT FOR EFFECTS OF APPLIED LOADS    NLDRV
C                     ON 1ST PARTS.W/RESP.TO STATE OF COSTATE EQS.          NLDRV
      LVDV   =  LVDV +  ASF1(AMVV,AVVV) - LGTV*(2.*(AG/V2 - AGV/V) +AGVV)  NLDRV
     *                                 -LPTCGV*(2.*(AP/V2 - APV/V)+APVV)  NLDRV
C                                                                           NLDRV
      LVDG   =  LVDG + LPTCGV*SGOCG * (AP/V - APV)                          NLDRV
C                                                                           NLDRV
      LVDR   =  LVDR +  ASF1(AMVR,AVVR) + LGTV*(AGR/V - AGVR)               NLDRV
     *               +  LPTCGV*(APR/V - APVR)                               NLDRV
C                                                                           NLDRV
      LVDT   =  LVDT +  ASF1(AMVZ,AVVZ) - LM*AMV - LV*AVV                   NLDRV
     *               +  LGTV*(AGZ/V - AGVZ) + LGAM/V  * (AG/V - AGV)        NLDRV
     *               +  LPSI/CGV * (APZ*TAUV - APVZ*TAU + AP/V - APV)       NLDRV
C                                                                           NLDRV
      LGDG   =  LGDG - LPTCGV*AP*(2./CG2 - 1.)                              NLDRV
C                                                                           NLDRV
      LGDR   =  LGDR -  LPTCGV*APR*SGOCG                                    NLDRV
C                                                                           NLDRV
      LGDT   =  LGDT -  SGOCG*(LPTCGV*APZ + LPSI*AP/CGV)                    NLDRV
C                                                                           NLDRV
      LRDR   =  LRDR +  ASF1(AMRR,AVRR) - LGTV*AGRR - LPTCGV*APRR           NLDRV
C                                                                           NLDRV
      LRDT   =  LRDT +  ASF1(AMRZ,AVRZ) - LM*AMR     - LV*AVR               NLDRV
     *               -  LGTV*AGRZ          - LGAM*AGR/V                     NLDRV
     *               -  LPTCGV*APRZ      - LPSI*APR/CGV                     NLDRV
      LMDM = ASF1(AMMM,AVMM) - LGTV*AGMM  - LPTCGV*APMM                     NLDRV
C                                                                           NLDRV
      LMDT = ASF1(AMMZ,AVMZ) - LM*AMM       - LV*AVM                        NLDRV
     *                         - LGTV*AGMZ  - LPTCGV*APMZ                   NLDRV
     *                         - LGAM*AGM/V - LPSI*APM/CGV                  NLDRV
C                                                                           NLDRV
      LTOT = ASF1(AMZZ,AVZZ) - LGTV*AGZZ  - LPTCGV*APZZ  - LM*AMZ           NLDRV
     *                         - 2.*(LV*AVZ + LGAM*AGZ/V + LPSI*APZ/CGV)    NLDRV
     *-LM*AMZ                                                               JUL19B
C                                                                           NLDRV
      LVDM = ASF1(AMVM,AVVM) + LGTV*(AGM/V - AGVM)                          NLDRV
     *                         + LPTCGV*(APM/V - APVM)                      NLDRV
C                                                                           NLDRV
      LGDM =-LPTCGV*SGOCG*APM                                               NLDRV
C                                                                           NLDRV
      LRDM = ASF1(AMRM,AVRM) - LGTV*AGRM  - LPTCGV*APRM                     NLDRV
C                                                                           NLDRV
C                                                                           NLDRV
C                                                                           NLDRV
C                                                                           NLDRV
      GO TO 5000                                                            NLDRV
C                    OPTIMAL CONTROL.  ACCOUNT FOR EFFECTS OF APPLIED       NLDRV
C                    LOADS ON 1ST PARTS.W/RESP.TO STATE OF COSTATE EQS.     NLDRV
 4500 LGDV = LVDG - LPTCGV*SGOCG*(APV - AP/V)                               NLDRV
      LPDV = LVDP                                                           NLDRV
      LPDG = LGDP                                                           NLDRV
      LRDV = LVDR + LVTAU* AVRV - LGTV*( AGRV - EAGR/V) - LPTCGV*           NLDRV
     *( APRV - EAPR/V) + LMTAU* AMRV                                        NLDRV
      LRDG = LGDR + LVTAU* AVRG - LGTV* AGRG - LPTCGV*( APRG + SGOCG*EAP    NLDRV
     *R)+LMTAU* AMRG                                                        NLDRV
      LRDP = LPDR                                                           NLDRV
      LODV = LVDO                                                           NLDRV
```

```
      LODG = LGDO                                                       NLDRV
      LODP = LPDO                                                       NLDRV
      LODR = LRDO                                                       NLDRV
      LMDV = TAU*(-LV* AVMV -(LGAM* AGMV + LPSI/COSGAM* APMV)/V - LM*   NLDRV
     * AMMV + (LGAM*EAGM + LPSI/COSGAM*EAPM)/V**2)                      NLDRV
      LMDG = TAU*(-LV* AVMG -(LGAM* AGMG + LPSI/COSGAM*( APMG + SGOCG*EA NLDRV
     *PM))/V- LM* AMMG)                                                 NLDRV
      LMDR = TAU*(-LV* AVMR -(LGAM* AGMR + LPSI/COSGAM* APMR)/V - LM*   NLDRV
     * AMMR)                                                            NLDRV
      LTDV = LVDT - LV*(AVV + TAU* AVZV) -LGAM/V*(AGV + TAU* AGZV - (AG NLDRV
     *+ EAGZ)/V) - LPSI/CGV*(APV + TAU* APZV - (AP + TAU*EAPZ)/V) -LM*  NLDRV
     *(AMV + TAU* AMZV)                                                 NLDRV
      LTDG = LGDT - LV*(AVG + TAU* AVZG) - LGAM/V*(AGG + TAU* AGZG) -    NLDRV
     *LPSI/CGV*(APG + TAU* APZG + SGOCG*(AP + TAU*EAPZ)) - LM*(AMG +    NLDRV
     *TAU* AMZG)                                                        NLDRV
      LTDP = LPDT                                                       NLCRV
      LTDR = LRDT - LV*(AVR + TAU* AVZR) -LGAM/V*(AGR + TAU* AGZR)      NLDRV
     * -LPSI/CGV*(APR + TAU* APZR) - LM*(AMR + TAU* AMZR)               NLDRV
      LTDD = LODT                                                       NLDRV
      LTDM = -LV*(AVM + TAU* AVZM) - LGAM/V*(AGM + TAU* AGZM) - LPSI/CGV NLDRV
     **(APM + TAU* APZM) - LM*(AMM + TAU* AMZM)                         NLDRV
      LVDV   =  LVDV   + ASF1( AMVV, AVVV) - LGTV*(2.*AG/V2  - (EAGV    NLDRV
     *                 + AGV)/V +  AGVV )  - LPTCGV*(2.*AP/V2           NLDRV
     *                 - (EAPV + APV)/V   +  APVV)                      NLDRV
C                                                                       NLDRV
      LVDG   =  LVDG   + ASF1( AMVG, AVVG) + LGTV*(AGG/V -  AGVG)       NLDRV
     *                 + LPTCGV*( SGOCG*(AP/V - EAPV) + APG/V -  APVG)  NLDRV
C                                                                       NLDRV
      LVDR   =  LVDR   + ASF1( AMVR, AVVR) + LGTV*(AGR/V -  AGVR)       NLDRV
     *                 + LPTCGV*(APR/V- APVR)                           NLDRV
C                                                                       NLDRV
      LVDT   =  LVDT   + ASF1( AMVZ, AVVZ) + LGTV*(AGZ/V -  AGVZ)       NLDRV
     *                 + LPTCGV*(APZ/V -  APVZ)                         NLDRV
     *                 - LV*EAVV - LM*EAMV + LGAM/V*(AG/V - EAGV)       NLDRV
     *                 + LPSI/CGV * (AP/V - EAPV)                       NLDRV
C                                                                       NLDRV
      LGDG   =  LGDG - LPTCGV*(AP*(2./CG2 - 1.) + SGOCG*APG)            NLDRV
C                                                                       NLDRV
      LGDR   =  LGDR  -LPTCGV*APR*SGOCG                                 NLDRV
C                                                                       NLDRV
      LGDT   =  LGDT  - LPSI/CGV*SGOCG*(AP + TAU*APZ)                   NLDRV
C                                                                       NLDRV
      LRDR   =  LRDR  + ASF1( AMRR, AVRR) - LGTV* AGRR - LPTCGV* APRR   NLDRV
C                                                                       NLDRV
      LRDT   =  LRDT  + ASF1( AMRZ, AVRZ) - LGTV* AGRZ - LPTCGV* APRZ   NLDRV
     *                - LV*EAVR - LM*EAMR - LGAM*EAGR/V - LPSI*EAPR/CGV NLDRV
      LVDM   =  ASF1( AMVM, AVVM)  + LGTV*(AGM/V -  AGVM)  +            NLDRV
     *          LPTCGV*(APM/V -  APVM)                                  NLDRV
C                                                                       NLDRV
      LGDM   =-LPTCGV*APM*SGOCG                                         NLDRV
C                                                                       NLDRV
      LRDM   =  ASF1( AMRM, AVRM)  - LGTV* AGRM  -  LPTCGV* APRM        NLDRV
C                                                                       NLDRV
      LMDM   =  ASF1( AMMM, AVMM)  - LGTV* AGMM  -  LPTCGV* APMM        NLDRV
C                                                                       NLDRV
      LMDT   =  ASF1( AMMZ, AVMZ)  - LGTV* AGMZ  -  LPTCGV* APMZ        NLDRV
     *        - LV*EAVM - LM*EAMM - LGAM*EAGM/V -  LPSI*EAPM/CGV        NLDRV
C                                                                       NLDRV
      LTDT = ASF1( AMZZ, AVZZ) - LGTV* AGZZ - LPTCGV* APZZ              NLDRV
     *       - LV*(EAVZ + AVZ) - LM*(EAMZ + AMZ) - LGAM*(EAGZ + AGZ)/V  NLDRV
     *       - LPSI*(EAPZ + APZ)/CGV                                    NLDRV
C                     STILL OPTIMAL CONTROL.  COMPUTE 1ST PARTS.W/RESP.TO NLDRV
C                     COSTATE OF STATE EQS.                             NLDRV
      VDLV =  TAU*AVLV                                                  NLDRV
      GDLV =  TAUV*AGLV                                                 NLDRV
      PDLV =  TAUCGV*APLV                                               NLDRV
      MDLV =  TAU*AMLV                                                  NLDRV
C                                                                       NLDRV
C                                                                       NLDRV
      VDLG  =  TAU*AVLG                                                 NLDRV
      GDLG  =  TAUV*AGLG                                                NLDRV
      PDLG  =  TAUCGV*APLG                                              NLDRV
      MDLG  =  TAU*AMLG                                                 NLDRV
C                                                                       NLDRV
      VDLP = TAU*AVLP                                                   NLDRV
```

```
      GDLP = TAUV*AGLP                                               NLDRV
      PDLP = TAUCGV*APLP                                             NLDRV
      MDLP = TAU*AMLP                                                NLDRV
C                                                                    NLDRV
      RETURN                                                         NLDRV
C                    NON-OPTIMAL CONTROL.  THE 1ST PARTS.W/RESP.TO   NLDRV
C                    COSTATE OF THE COSTATE EQS. ARE GIVEN BY NEGATIVE NLDRV
C                    TRANSPOSE OF THE 1ST PARTS.W/RESP.TO STATE OF THE NLDRV
C                    STATE EQS.                                      NLDRV
 5000 CONTINUE                                                       NLDRV
      DO 10 I = 1, 9                                                 NLDRV
      K = I + 9                                                      NLDRV
      DO 10 J = 1, 9                                                 NLDRV
   10 JACOB(J + 9, K) = -JACOB(I, J)                                 NLDRV
C                    ALSO FOR NON-OPTIMAL CONTROL, THE 1ST PARTS.W/RESP. NLDRV
C                    TO STATE OF THE COSTATE EQS. FORM A SYMMETRIC   NLDRV
C                    MATRIX, SO FILL IN LOWER TRIANGULAR PORTION.    NLDRV
 6000 CONTINUE                                                       NLDRV
      DO 20 I = 2, 9                                                 NLDRV
      L = I + 9                                                      NLDRV
      K = I-1                                                        NLDRV
      DO 20 J = 1,K                                                  NLDRV
   20 JACOB(L, J) = JACOB(J + 9, I)                                  NLDRV
C                                                                    NLDRV
C                                                                    NLDRV
C                                                                    NLDRV
C                                                                    NLDRV
      RETURN                                                         NLDRV
      END                                                            NLDRV
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AE	A_{exit}	I	Total nozzle exit area (FT^2)	/DYNA	/(89)	APPLY	I	AE
						ARCIN	O	AE
						IMPULS	I	AE
						NLDRV	I	AE
						TH2	I	AE
APLP	$\partial a^v/\partial\lambda_v$	I	The first entry in a 4x3 matrix that contains $\partial a/\partial\lambda_v$, $\partial a/\partial\lambda_\gamma$ and $\partial a/\partial\lambda_\psi$	/AXLE	/(335)	APPLY	M	APLP
						NLDRV	I	APLP
AV	a^v	I	The first word of a four word array that contains the vector of applied accelerations defined by Equation 2.4-2 in Vol.I of this document.	/AXLE	/(1)	APPLY	M	AV
						APPLY	O	AXLE
						NLDRV	I	AV
AVV	$\partial a^v/\partial v$	I	The first entry in a 4x8 matrix that contains the total partial derivatives of the a vector with respect to the QL state vector (excluding the heating state), $\partial a/\partial y$.	/AXLE	/(5)	APPLY	M	AVV
						NLDRV	I	AVV
AVVG	$\partial(\delta a^v/\delta V)/\partial\gamma$	I	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\gamma$	/AXLE	/(101)	NLDRV	I	AVVG
AVVP	$\partial(\delta a^v/\delta V)/\partial\psi$	I	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial\psi$	/AXLE	/(133)	NLDRV	I	AVVP
AVVR	$\partial(\delta a^v/\delta V)/\partial h$	I	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial h$	/AXLE	/(165)	APPLY	O	AVVR
						NLDRV	I	AVVR
AVVV	$\partial(\delta a^v/\delta V)/\partial V$	I	The first entry in a 4x8 matrix that contains $\partial(\delta a/\delta y)/\partial V$.	/AXLE	/(69)	APPLY	O	AVVV
						APPLY	I	AYY
						NLDRV	I	AVVV
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D	/(11)	BNDRY	I	C
						BRANPT	I	C
						GROPE	I	C
						INTRPT	I	C
						NEWCS	M	C
						NLDRV	I	C
						NORMAL	I	C
						WRAPUP	I	C
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol	/DYNA	/(151)	AL1	I	CODAE
						AL4	I	CODAE
						AL6	I	CODAE
						AL7	I	CODAE
						AL8	I	CODAE
						AL9	I	CODAE
						APPLY	I	CODAE
						CONTRL	I	CODAE
						NLDRV	I	CODAE
						TH3	I	CODAE
						UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol	/DYNA	/(10)	AL1	I	COSA
						AL4	I	COSA
						AL6	I	COSA
						AL7	I	COSA
						AL8	I	COSA
						AL9	I	COSA
						APPLY	I	COSA
						CONTRL	I	COSA
						NLDRV	I	COSA
						OUTPUT	I	COSA
						TH3	I	COSA
						UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol	/DYNA	/(4)	AL1	I	COSGAM
						AL4	I	COSGAM
						AL7	I	COSGAM
						AL8	I	COSGAM
						AL9	I	COSGAM
						CONTRL	I	COSGAM
						NLDRV	I	COSGAM
						OUTPUT	I	COSGAM
						PDBCQL	I	COSGAM
						STATEF	M	COSGAM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
COSPSI	$\cos\psi$	I	See symbol	/DYNA /(	95)	AL4 AL7 AL8 AL9 CONTRL NLDRV PDBCOL STATEF	I I I I I I I O	COSPSI COSPSI COSPSI COSPSI COSPSI COSPSI COSPSI COSPSI
COSRHO	$\cos\rho$	I	See symbol	/DYNA /(	97)	AL4 AL7 AL8 AL9 CONTRL NLDRV OUTPUT PDBCOL STATEF	I I I I I I I I M	COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO COSRHO
COS2RO	$\cos 2\rho$	I	See symbol	/DYNA /(	120)	AL4 AL7 AL8 NLDRV STATEF	I I I I O	COS2RO COS2RO COS2RO COS2RO COS2RO
DB	D_b	I	Base drag (LBS)	/DYNA /(	163)	AL1 AL4 AL6 AL7 AL8 AL9 APPLY CONTRL NLDRV OUTPUT STATEF TH3 UT	I I I I I I I I I I I I I	DB DB DB DB DB DB DB DB DB DB DB DB DB
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA /(	69)	AL5 AL7 AL8 AL9 APPLY CONTRL ENVPRO NLDRV OUTPUT TH3 UT	I I I I I I I I I I M	DRAG DRAG DRAG DRAG DRAG DRAG DRAG DRAG DRAG DRAG DRAG
EAVV	$\delta a^v/\delta v$	I	The first entry in a 4x8 matrix that contains, when α is optimal, $\delta a/\delta y = \partial a/\partial y\|_\alpha = \text{constant}$ When α is nonoptimal, $\delta a/\delta y = \partial a/\partial y$	/AXLE /(	37)	APPLY NLDRV	I I	EAVV EAVV
FVAC		I	Total vacuum thrust (rocket) (LBS)	/DYNA /(	33)	APPLY ARCIN IMPULS NLDRV STATEF TH2	I M M I M I	FVAC FVAC FVAC FVAC FVAC FVAC
G	g	I	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA /(	8)	AL4 AL7 AL8 AL9 CONTRL NLDRV STATEF	I I I I I I M	G G G G G G G
GAMMAD		I	Pitch rate (RAD/SEC)	/DYNA /(	88)	AL4 ARCIN CONTRL NLDRV	I O I I	GAMMAD GAMMAD GAMMAD GAMMAD

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GH	$\partial g/\partial R$	I	See symbol	/DYNA	/(142)	AL7 AL8 NLDRV STATEF	I I I M	GH GH GH GH
GRR	$\partial^2 g/\partial R^2$	I	See symbol	/DYNA	/(143)	AL7 AL8 NLDRV STATEF	I I I M	GRR GRR GRR GRR
IATM		I	Atmosphere option flag	/ARCDAT	/(7)	ARCIN NLDRV OUTPUT STATEF	I I I I	IATM IATM IATM IATM
JACOB		M	An 18x18 array defined by Equation 17.5-5 in Vol.I of this document. The entry in the i-th row and j-th column is the total partial derivative of the quasitime derivative of the Y_i component of Y with respect to the Y_j component of Y, i.e., $\partial Y_i'/\partial Y_j$, where $Y^T = (y^T, \lambda^T)$	/JACOB	/(1)	LINDRV NLDRV NLDRV SALVE	I M M O	JAKE JACOB VDV JAKE
KODE		I	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; *KODE = 5: α nonoptimal and $\phi = 0$.*	/DYNA	/(25)	APPLY ARCIN CONTRL FORCES NLDRV STATEF	I O M I I I	KODE KODE KODE KODE KODE KODE
KODES		M	Not used.	/CNTRL	/(56)	GROPE NLDRV WRAPUP	O M O	KODES KODES KODES
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL	/(28)	ALGCON APPLY ARCIN COMOMO GROPE NLDRV OUTPUT RKUTT1	I I I O O I I I	KONVER KONVER KONVER KONVER KONVER KONVER KONVER KONVER
LGAM	λ_γ	I	Relative flight path angle costate	/D	/(101)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LGAM LGAM LGAM LGAM LGAM LGAM
LHT	λ_Q	I	Heating costate	/D	/(108)	NLDRV WRAPUP	I I	LHT LHT
LM	λ_m	I	Mass costate	/D	/(106)	NLDRV OUTPUT WRAPUP	I I I	LM LM LM
LMU	λ_μ	I	Relative longitude costate	/D	/(105)	NLDRV OUTPUT WRAPUP	I I I	LMU LMU LMU
LPSI	λ_ψ	I	Relative azimuth angle costate	/D	/(102)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LPSI LPSI LPSI LPSI LPSI LPSI
LR	λ_R	I	Altitude costate	/D	/(103)	NLDRV OUTPUT WRAPUP	I I I	LR LR LR
LRHO	λ_ρ	I	Latitude costate	/D	/(104)	NLDRV OUTPUT WRAPUP	I I I	LRHO LRHO LRHO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LV	λ_v	I	Relative velocity costate	/D	/(100)	AL1	I	LV
						CONTRL	I	LV
						NLDRV	I	LV
						OUTPUT	I	LV
						WRAPUP	I	LV
M	m	I	Mass (G'S)	/D	/(97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
N		I	Total number of QL state and costate variables. N = 18.	/PC	/(2)	BNDRY	I	N
						CHECK	I	N
						INARC	I	N
						LINDRV	I	N
						NLDRV	I	N
						NORMAL	I	N
						RKUTTI	I	N
						SALVE	I	N
						WRAPUP	I	N
NOM	V	O	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
OMEGAT	2ω	I	See symbol	/DYNA	/(12)	NLDRV	I	OMEGAT
						TRAJIN	O	OMEGAT
OMEGA2	ω^2	I	See symbol	/DYNA	/(6)	AL4	I	OMEGA2
						AL7	I	OMEGA2
						AL8	I	OMEGA2
						AL9	I	OMEGA2
						NLDRV	I	OMEGA2
						TRAJIN	O	OMEGA2
PA	P_a	I	Atmospheric pressure (LBS/FT^2)	/DYNA	/(14)	IMPULS	I	PA
						NLDRV	I	PA
						OUTPUT	I	PA
						TH2	I	PA
QMULT	=0 OR 1	I	Heating flag multiplier	/ARCDAT/(	40)	NLDRV	I	QMULT
						PDBCOL	I	QMULT

1169

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRQ	I	R
						NLDRV	I	R
						PDBCOL	I	R
						DLTDSZ	I	R
						STATEF	M	R
RHOB	ρ_b	I	Atmosphere base density for heating calculation (LB/FT**3)	/ARCDAT	/(39)	NLDRV	I	RHOB
						PDBCOL	I	RHOB
RO	ρ_a	I	Atmospheric density (SLGS/FT^3)	/DYNA	/(15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCOL	I	RO
						STATEF	I	RO
ROR	$\partial\rho_a/\partial R$	I	See symbol	/DYNA	/(19)	AL7	I	ROR
						AL8	I	ROR
						AL9	I	ROR
						NLDRV	I	ROR
						PDBCOL	I	ROR
						STATEF	I	ROR
RORR	$\partial^2\rho_a/\partial R^2$	I	See symbol	/DYNA	/(23)	AL7	I	RORR
						AL8	I	RORR
						AL9	I	RORR
						NLDRV	I	RORR
						STATEF	I	RORR
SINGAM	$\sin\gamma$	I	See symbol	/DYNA	/(3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM
SINPSI	$\sin\psi$	I	See symbol	/DYNA	/(94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCOL	I	SINPSI
						STATEF	O	SINPSI
SINRHO	$\sin\rho$	I	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCOL	I	SINRHO
						STATEF	M	SINRHO
SIN2RO	$\sin 2\rho$	I	See symbol	/DYNA	/(119)	AL4	I	SIN2RO
						AL7	I	SIN2RO
						AL8	I	SIN2RO
						NLDRV	I	SIN2RO
						STATEF	M	SIN2RO
TAU	τ	I	Subarc duration (SEC)	/D	/(98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
VDV		M	An 18x18 array defined by Equation 17.5-5 in Vol.I of this document. The entry in the i-th row and j-th column is the total partial derivative of the quasitime derivative of the Y_i component of Y with respect to the Y_j component of Y, i.e., $\partial Y_i' / \partial Y_j$, where $Y^T = (y^T, \lambda^T)$	/JACOB	/(1)	LINDRV	I	JAKE
						NLDRV	M	JACOB
						NLDRV	M	VDV
						SALVE	O	JAKE

SUBRØUTINE NØMNAL

1172

Purpose

NOMNAL combines up the particular and homogeneous solutions resulting from the preceding iteration to get the total solution from the preceding iteration,

$$s = p + Hc.*$$

*See Sections 17.1 and 17.2 of Vol. I.

NOMNAL

```
      SUBROUTINE NOMNAL                                                 NOMNAL
C                                                                       NOMNAL
C                    THIS ROUTINE COMBINES UP THE PARTICULAR AND HOMO-  NOMNAL
C                    GENEOUS SOLUTIONS TO YIELD THE TOTAL SOLUTION.     NOMNAL
C                                                                       NOMNAL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /S/ S                                                      NOMNAL
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /Z/ Z(50)                                                  Z
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      DIMENSION S(820)                                                  NOMNAL
      IF(NN - N) 1, 1, 2                                                NOMNAL
    1 READ(ITAPA) (Z(IK), IK = 1, N)                                    NOMNAL
      RETURN                                                            NOMNAL
    2 READ(ITAPA) (S(IK), IK = 1, NN)                                   NOMNAL
      CALL MATMLT(Z, S(N + 1), C, N, MOM, 1)                            NOMNAL
      CALL MATADD(Z, Z, S, N, 1)                                        NOMNAL
      RETURN                                                            NOMNAL
      END                                                               NOMNAL
```

1174

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.	/D	/(11)	BNDRY BRANPT GROPE INTRPT NEWCS NLDRV NOMNAL WRAPUP	I I I I M I I I	C C C C C C C C
ITAPA		I	Number of the logical unit containing the quasitime histories of the particular and homogeneous solutions from the preceding QL iteration.	/CNTRL	/(3)	GROPE NOMNAL	O I	ITAPA ITAPA
MOM		I	The number of homogeneous solutions currently being integrated.	/CNTRL	/(9)	GROPE INARC LINDRV NOMNAL SALVE WRAPUP	O M I I M M	MOM MOM MOM MOM MOM MOM
N		I	Total number of QL state and costate variables. N = 18.	/PC	/(2)	BNDRY CHECK INARC LINDRV NLDRV NOMNAL RKUTT1 SALVE WRAPUP	I I I I I I I I I	N N N N N N N N N
NN		I	The number of quantities currently being numerically integrated.	/CNTRL	/(52)	BNDRY INARC MADAMS MAGIC NOMNAL RKUTT1 RKUTT2 SALVE WRAPUP	M M I M I I I M M	NN NN NN NN NN NN NN NN NN
S		M	An 820 word array used to store the particular and homogeneous solutions.	/S	/(1)	NOMNAL RKUTT1	M M	S S
Z	Z	M	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY BRANPT ENDPT ENVPRQ INTERP INTRPT LINDRV NOMNAL OUTPUT RKUTT1 RKUTT2 SALVE WRAPUP	I I I I O I I M I O M M M	Z Z Z Z Z Z Z Z Z Z Z Z Z

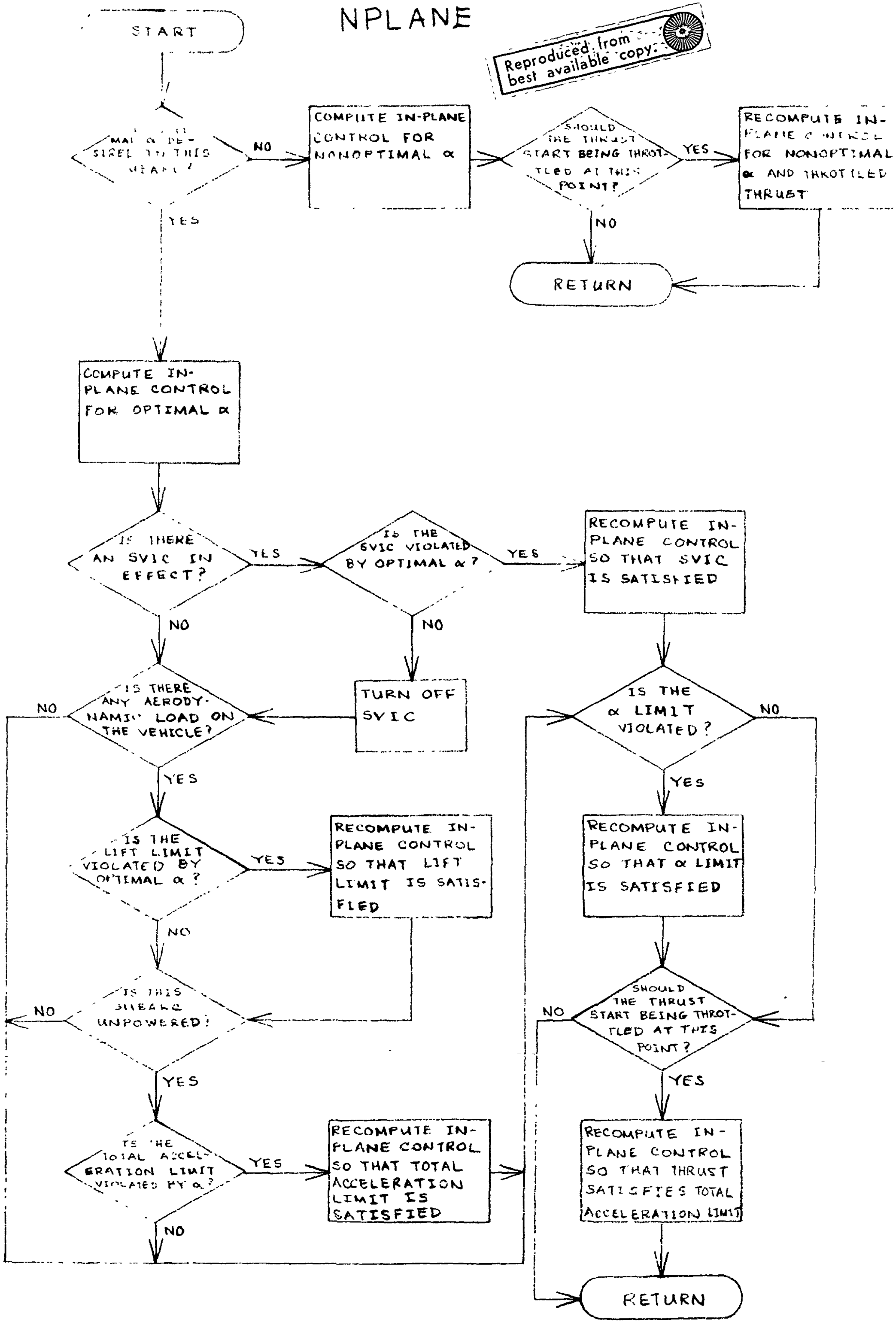
SUBRØUTINE
NPLANE

1176

Purpose

NPLANE controls the calculation of the in-plane control vector.

NPLANE


START
NO
YES
COMPUTE IN-PLANE CONTROL FOR NONOPTIMAL α
SHOULD THE THRUST START BEING THROTTLED AT THIS POINT?
YES
NO
RECOMPUTE IN-PLANE CONTROL FOR NONOPTIMAL α AND THROTTLED THRUST
RETURN
COMPUTE IN-PLANE CONTROL FOR OPTIMAL α
IS THERE AN SVIC IN EFFECT?
YES
NO
IS THE SVIC VIOLATED BY OPTIMAL α?
YES
NO
RECOMPUTE IN-PLANE CONTROL SO THAT SVIC IS SATISFIED
TURN OFF SVIC
IS THERE ANY AERODYNAMIC LOAD ON THE VEHICLE?
NO
YES
IS THE α LIMIT VIOLATED?
NO
YES
IS THE LIFT LIMIT VIOLATED BY OPTIMAL α?
YES
NO
RECOMPUTE IN-PLANE CONTROL SO THAT LIFT LIMIT IS SATISFIED
RECOMPUTE IN-PLANE CONTROL SO THAT α LIMIT IS SATISFIED
UNPOWERED?
NO
YES
SHOULD THE THRUST START BEING THROTTLED AT THIS POINT?
NO
YES
TOTAL ACCELERATION LIMIT VIOLATED BY α?
YES
NO
RECOMPUTE IN-PLANE CONTROL SO THAT TOTAL ACCELERATION LIMIT IS SATISFIED
RECOMPUTE IN-PLANE CONTROL SO THAT THRUST SATISFIES TOTAL ACCELERATION LIMIT
RETURN

```
      SUBROUTINE NPLANE                                                 NPLANE
C                                                                       NPLANE
C                 THIS ROUTINE CONTROLS THE CALCULATION OF THE IN-      NPLANE
C                 PLANE CONTROL QUANTITIES T, DELTAE, AND ALPHA.        NPLANE
C                                                                       NPLANE
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF       ,EJ         ,XISP       ,TMULT      ,DTNC       ,DTPI       ,  ARCDAT
     *IATM       ,IMODE      ,JAER       ,JPRO       ,QMAX       ,GMAX       ,  ARCDAT
     *XLMAX      ,HDMAX      ,GMDOT      ,ALFMAX     ,PHMAX      ,MAEA       ,  ARCDAT
     *MAEB       ,MAEC       ,MAED       ,MAEE       ,MAEF       ,MAEG       ,  ARCDAT
     *MT         ,MISP       ,MXCG       ,MZCG       ,MWDA       ,MWDB       ,  ARCDAT
     *MDB        ,XCGR       ,ZCGR       ,XE         ,ZE         ,XT         ,  ARCDAT
     *DREF       ,MCND       ,RHOB       ,QMULT      ,REMAX      ,FRATE         ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,COE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PRODI(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TQ ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT      MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
```

```
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2               MATS
      COMMON /MATS/                                                      MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)   MATS
      COMMON /MATS/                                                      MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,     MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,     MATS
     *POR   ,POO   ,PLG   ,PLP                                           MATS
      EQUIVALENCE(PROD1,PROD5)                                           MATS
      DATA RAD/.01745329251994 33/                                       NPLANE
C                       IS THE DESIRED CONTROL OPTIMAL.                  NPLANE
      IF(IMODE .LE. 2) GO TO 102                                         NPLANE
C                       THE DESIRED CONTROL IS NON-OPTIMAL               NPLANE
  101 CALL ALGCON(J1, J2, J3)                                            NPLANE
C                       CHECK TO SEE IF SHOULD BE THROTTLING             NPLANE
      CALL THROTL                                                        NPLANE
C                       IF JUST EXCEEDED TOTAL ACCEL. LIMIT, RECOMPUTE   NPLANE
C                       IN-PLANE CONTROL.                                NPLANE
      IF(SWITCH) CALL ALGCON(J1, J2, J3)                                 NPLANE
      RETURN                                                             NPLANE
C                       DESIRED CONTROL IS OPTIMAL.  CALC. IN-PLANE CONTROL NPLANE
C                       USING OPTIMAL ANGLE OF ATTACK                    NPLANE
  102 J3 = 1                                                             NPLANE
  103 CALL ALGCON(J1, J2, J3)                                            NPLANE
C                       ARE THERE ANY STATE VARIABLE INEQUALITY CONSTRAINTS NPLANE
C                       IN EFFECT.                                       NPLANE
      IF(INQF .EQ. 0) GO TO 109                                          NPLANE
C                       YES, SVIC IN EFFECT.  FIND OUT WHICH ONE         NPLANE
      IF(INQF - 8) 104, 105, 106                                         NPLANE
C                       DYNAMIC PRESSURE                                 NPLANE
  104 CALL AL7000                                                        NPLANE
      GO TO 107                                                          NPLANE
C                       REYNOLDS NUMBER                                  NPLANE
  105 CALL AL8000                                                        NPLANE
      GO TO 107                                                          NPLANE
  106 CALL AL9000                                                        NPLANE
C                       WAS SVIC VIOLATED                                NPLANE
  107 IF(XK3 .GT. 0.) GO TO 108                                          NPLANE
C                       NO.  TURN OFF SVIC.                              NPLANE
      INQF = 0                                                           NPLANE
      IDAM = 0                                                           NPLANE
      GO TO 109                                                          NPLANE
C                       SVIC VIOLATED. RECOMPUTE IN-PLANE CONTROL ON SVIC. NPLANE
  108 J3 = INQF                                                          NPLANE
      CALL ALGCON(J1, J2, J3)                                            NPLANE
      GO TO 111                                                          NPLANE
C                       SVIC NOT IN EFFECT.  IS THERE ANY AERODYNAMIC LOAD NPLANE
C                       ON THE VEHICLE                                   NPLANE
  109 IF(.NOT.ILOAD) GO TO 111                                           NPLANE
C                       YES.  CHECK FOR LIFT LIMIT VIOLATION.            NPLANE
      IF(XLMAX .LE. 0. .OR. ABS(ULFT) .LE. XLMAX) GO TO 110              NPLANE
C                       LIFT LIMIT VIOLATED.  PUT CONTROL ON LIMIT.      NPLANE
      CULFT = SIGN(XLMAX, ULFT)                                          NPLANE
      J3 = 3                                                             NPLANE
      CALL ALGCON(J1, J2, J3)                                            NPLANE
C                       IS THIS POWERED FLIGHT                           NPLANE
  110 IF(IPOW .NE. 0) GO TO 111                                          NPLANE
C                       UNPOWERED.  CHECK FOR TOTAL ACCEL. LIMIT VIOLATION. NPLANE
      IF(GMAX .LE. 0.) GO TO 111                                         NPLANE
      CALL AL5000                                                        NPLANE
      IF(XK3 .LE. 0.) GO TO 111                                          NPLANE
C                       UNPOWERED TOTAL ACCEL. LIMIT VIOLATION.  PUT CONTROL NPLANE
C                       ON LIMIT.                                        NPLANE
      J3 = 5                                                             NPLANE
      CALL ALGCON(J1, J2, J3)                                            NPLANE
C                       CHECK FOR ALPHA LIMIT VIOLATION.                 NPLANE
```

```
  111 IF(ALFMAX .LE. 0. .OR. ABS(ALPHA) .LE  ALMAX) GO TO 112          NPLANE
C               ALPHA LIMIT VIOLATED.  PUT CONTROL ON LIMIT.           NPLANE
      CALPHA = SIGN(ALMAX, ALPHA)                                      NPLANE
      INQF = 0                                                         NPLANE
      J3 = 2                                                           NPLANE
      CALL ALGCON(J1, J2, J3)                                          NPLANE
C               CHECK FOR POWERED ACCEL. LIMIT.                        NPLANE
  112 CALL THROTL                                                      NPLANE
      IF(SWITCH) CALL ALGCON(J1, J2, J3)                               NPLANE
      RETURN                                                           NPLANE
      END                                                              NPLANE
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ALFMAX	α_{MAX}	I	Maximum angle of attack (DEG)	/ARCDAT/(	16)	ARCIN NPLANE	I I	ALFMAX ALFMAX
ALMAX	α_{max}	I	Magnitude of angle of attack constarint (RADS)	/DYNA /(	162)	ARCIN NPLANE	O I	ALMAX ALMAX
ALPHA	α	I	Angle of attack (RAD)	/DYNA /(	79)	AEROCO ALGCON AL2 ARCIN CONTRL ENVPRO MOMECO NPLANE OUTPUT TRAJIN UT WRAPUP	I M I M M I I I I O I I	ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA
CALPHA	C_{α}	O	Value for angle of attack in case constant angle of attack constraint is used. (RADS)	/DYNA /(	161)	AL2 NPLANE	I O	CALPHA CALPHA
CULFT	C_{L_u}	O	Magnitude of untrimmed lift limit (LBS)	/DYNA /(	165)	AL3 NPLANE	I O	CULFT CULFT
GMAX	G_{MAX}	I	Maximum total acceleration g load	/ARCDAT/(	12)	AL5 NPLANE THROTL TH3	I I I I	GMAX GMAX GMAX GMAX
IDAM		*O*	*Optional atmospheric calculations flag.* IDAM = -1: Compute $\partial^3\rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3\rho_a/\partial R^3$, μ_a, $\partial\mu_a/\partial R$, etc.	*/DYNA /(*	*218)*	*ARCIN* ERROR NPLANE STATEF WRAPUP	*O* I O I O	*IDAM* IDAM IDAM IDAM IDAM
ILOAD		I	Logical flag that is true if there is any aerodynamic load on the vehicle.	/DYNA /(	181)	ARCIN CONTRL NPLANE UT	M I I I	ILOAD ILOAD ILOAD ILOAD
IMODE		I	Control mode option flag	/ARCDAT/(	8)	ARCIN CONTRL NPLANE	I I I	IMODE IMODE IMODE
INQF		M	State variable inequality constraint flag. INQF = 0: No SVIC in effect; INQF = 7: Dynamic pressure IC in effect; INQF = 8: Heating rate SVIC in effect; INQF = 9: Reynolds number SVIC in effect.	/DYNA /(	185)	ARCIN NPLANE	M M	INQF INQF
IPOW		I	Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag	/DYNA /(	139)	ARCIN FORCES NPLANE STATEF THROTL	M I I I I	IPOW IPOW IPOW IPOW IPOW
J1		I	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY ARCIN CONTRL FORCES NPLANE STATEF THROTL	I O M I I I M	J1 J1 J1 J1 J1 J1 J1
J2		I	Engine deflection option flag. J2 = 1: Constant engine deflection; J2 = 2: Moment balancing.	/DYNA /(	174)	ARCIN CONTRL NPLANE	O I I	J2 J2 J2
J3		M	Angle of attack option flag. J3 = 1: Optimal angle of attack; J3 = 2: Constant angle of attack; J3 = 3: Untrimmed lift limit; J3 = 4: Vertical rise or pitchover; J3 = 5: Unpowered total acceleration limit; J3 = 6: Gravity turn; J3 = 7: Dynamic pressure constraint; J3 = 8: Heating rate constraint; J3 = 9: Reynolds number constraint.	/DYNA /(	175)	ARCIN CONTRL NPLANE OUTPUT	O M M I	J3 J3 J3 J3

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SWITCH		I	Logical flag that is true if this is the compute point at which the powered acceleration constraint commences.		/DYNA /(	184)	CONTRL NPLANE THROTL	I I O	SWITCH SWITCH SWITCH
ULFT	L_u	I	Untrimmed aerodynamic lift	(LBS)	/DYNA /(	164)	AL3 NPLANE UT	I I M	ULFT ULFT ULFT
XLMAX	L_{MAX}	I	Maximum aerodynamic lift	(LBS)	/ARCDAT/(	13)	NPLANE	I	XLMAX

SUBRØUTINE ØUTPUT

Purpose

OUTPUT puts print quantities into the print array AP and controls the computation of auxiliary print quantities.

```
      SUBROUTINE OUTPUT                                                 OUTPUT
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTMR,ULFTRA,IPOW  ,XARE  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /ZD/ ZD(50)                                                ZD
      COMMON /Z/ Z(50)                                                  Z
      DIMENSION PRODI(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
```

```
      *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,     MATS
      *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,     MATS
      *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
      *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
      *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
      *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
      *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
      *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
      *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2               MATS
       COMMON /MATS/                                                       MATS
      *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)    MATS
       COMMON /MATS/                                                       MATS
      *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,     MATS
      *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,     MATS
      *POR   ,POO   ,PLG   ,PLP                                            MATS
       EQUIVALENCE(PROD1,PROD5)                                            MATS·
       COMMON/ARCDAT/                                                      ARCDAT
      *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
      *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
      *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
      *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
      *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      , ARCDAT
      *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
      *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
       DIMENSION  ARCDA(40)                                                ARCDAT
       EQUIVALENCE(SREF,ARCDA)                                             ARCDAT
       COMMON/GLOBAL/                                                      GLOBAL
      *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,     GLOBAL
      *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20),   GLOBAL
      *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,         GLOBAL
      *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                            GLOBAL
       COMMON/ ORBIT/  VI,       GAMI,      PSII,      XMUI,        P,     ORBIT
      *     ECC,      AINCL,      ARGP,    ASCNOD,    SMIMAJ,   APOGEE,    ORBIT
      *  PERGEE,     ANOMLY,      CAPX,      CAPY,     ASYMP,   ENERGY,    ORBIT
      *   HMNTM,     DVIDV,     DVIDG,                                     ORBIT
      *  DVIDH ,     DVIDM,     DVIDPS,    DVIDRO,    DVIDMU,    DGIDV,    ORBIT
      *  DGIDG ,     DGIDH,     DGIDM ,    DGIDPS,    DGIDRO,    DGIDMU,   ORBIT
      *  DPIDV ,     DPIDG,     DPIDH ,    DPIDM ,    DPIDPS,    DPIDRO,   ORBIT
      *  DPIDMU,     DMIDV,     DMIDG ,    DMIDH ,    DMIDM ,    DMIDPS,   ORBIT
      *  DMIDRO,     DMIDMU,    DPDV  ,    DPDG  ,    DPDH  ,    DPDM  ,   ORBIT
      *  DPDPS ,     DPDRO ,    DPDMU ,    DECDV ,    DECDG ,    DECDH ,   ORBIT
      *  DECDM ,     DECDPS,    DECDRO,    DECDMU,    DIDV  ,    DIDG  ,   ORBIT
      *  DIDH  ,     DIDM  ,   DIDPS  ,   DIDRO  ,   DIDMU  ,   DBEDV  ,   ORBIT
      *  DBEDG ,     DBEDH ,    DBEDM ,    DBEDPS,    DBEDRO,    DBEDMU,   ORBIT
      *  DNODV ,     DNODG ,    DNODH ,    DNODM ,    DNODPS,    DNODRO,   ORBIT
      *  DNODMU,     DSMDV,      DSMDG,      DSMDH,     DSMDM,   DSMDPS     ORBIT
       COMMON/ORBIT/                                                       ORBIT
      *  DSMDRO,    DSMDMU,     DAPDV,     DAPDG,      DAPDH,    DAPDM ,   ORBIT
      *  DAPDPS,    DAPDRO,     DAPDMU,    DPEDV,      DPEDG,    DPEDH ,   ORBIT
      *DPEDM     ,DPEDPS     ,DPEDRO     ,DPEDMU     ,DANDV     ,DANDG   , ORBIT
      *  DANDH ,    DANDM ,     DANDPS,    DANDRO,    DANDMU,    DCXDV ,   ORBIT
      *  DCXDG ,    DCXDH ,     DCXDM ,    DCXDPS,    DCXDRO,    DCXDMU,   ORBIT
      *  DCYDV ,    DCYDG ,     DCYDH ,    DCYDM ,    DCYDPS,    DCYDRO,   ORBIT
      *  DCYDMU,    DASDV ,     DASDG ,    DASDH ,    DASDM ,    DASDPS,   ORBIT
      *  DASDRO,    DASDMU,     DENDV ,    DENDG ,    DENDH ,    DENDM ,   ORBIT
      *  DENDPS,    DENDRO,     DENDMU,    DMODV ,    DMODG ,    DMODH ,   ORBIT
      *  DMODM ,    DMODPS,     DMODRO,    DMODMU                          ORBIT
       DIMENSION ORBPRM(18),PPO(7,18)                                      ORBIT
       EQUIVALENCE (VI,ORBPRM),(DVIDV,PPO)                                 ORBIT
       COMMON/ORBIT/ YMXRF,SNXLMR,CSXLMR,SDOWN,SCROSS,TD,TC                ORBIT
      *,SNPSR     ,CSPSR     ,SNGI      ,CSGI      ,SPSII     ,CPSII     , ORBIT
      * STOT      ,CSI       ,SNI       ,SNGNU     ,CSAND     ,COSDMU    , ORBIT
      * SINDMU    ,THT       ,WTFUEL                                       ORBIT
       COMMON / PRINT/ AP(100)                                             OUTPUT
       EQUIVALENCE (VNU, DYN214)                                           OUTPUT
       DIMENSION FF(9)                                                     OUTPUT
       DATA AP / 100* 03777777777777777777/                                OUTPUT
       DATA RAD/57.2957795130823/, PI/3.14159265358979/                    OUTPUT
       FTNM = 1./6076.10333                                                OUTPUT
       GAMD= GAM*RAD                                                       OUTPUT
       PSIDG=PSI*RAD                                                       OUTPUT
       AP(1)= TIME                                                         OUTPUT
       AP(2)= ALT                                                          OUTPUT
       AP(3)= V                                                            OUTPUT
       AP(4)=GAMD                                                          OUTPUT
```

```
      AP(5)=PSIDG                                                  OUTPUT
      AP(6)= MU*RAD                                                OUTPUT
      AP(7)= RHO*RAD                                               OUTPUT
      AP(8) = TAU*XX                                               OUTPUT
      AP(9)= W                                                     OUTPUT
      MO=0                                                         OUTPUT
      CALL PDBCQL(-1,FF,FF,FF,0,MO)                                OUTPUT
      AP(10) = VI                                                  OUTPUT
      AP(11) = GAMI*RAD                                            OUTPUT
      AP(12) = PSII*RAD                                            OUTPUT
      AP(13) = XMUI*RAD                                            OUTPUT
      AP(14) = SCROSS*FTNM                                         OUTPUT
      AP(16)= M                                                    OUTPUT
      AP(18)= MT                                                   OUTPUT
      AP(19) = ZD(9)                                               OUTPUT
      AP(20) = STOT *FTNM                                          OUTPUT
      AP(21) = SDOWN*FTNM                                          OUTPUT
      IF(IATM .LT. 2) RE = V*RO/VMU                                OUTPUT
      AP(22)= RE                                                   OUTPUT
      AP(23)= PA                                                   OUTPUT
      AP(24)= RO                                                   OUTPUT
      AP(25)= CS                                                   OUTPUT
      AP(26)=      MACH                                            OUTPUT
      AP(45) = J3                                                  OUTPUT
      AP(46)= XK3                                                  OUTPUT
      AP(29) = ALPHA*RAD                                           OUTPUT
      AP(30) = PHI*RAD                                             OUTPUT
      SACHI = SINA*COSPHI                                          OUTPUT
      IF(ABS(SACHI).GT.1.) SACHI=SIGN(1.,SACHI)                    OUTPUT
      SASHI=SINA*SINPHI                                            OUTPUT
      IF(ABS(SASHI).GT.1.) SASHI=SIGN(1.,SASHI)                    OUTPUT
      AP(92)= GAMD + ASIN(SACHI)*RAD                               OUTPUT
      AP(93) = PSIDG + ASIN(SASHI)*RAD                             OUTPUT
      AP(31)= XJ                                                   OUTPUT
      AP(32)= Q                                                    OUTPUT
      AP(33)= XMCG                                                 OUTPUT
      AP(34)= LIFT                                                 OUTPUT
      AP(35)= DRAG                                                 OUTPUT
      AP(36)= T                                                    OUTPUT
      IF(ZD(7) .EQ. 0.) GO TO 10                                   OUTPUT  10
      AP(43) = -T/ZD(7)/GR*TAU                                     OUTPUT
      GO TO 20                                                     OUTPUT  20
   10 AP(43)=0.                                                    OUTPUT
   20 CONTINUE                                                     OUTPUT
      AP(48)= DELTAE *RAD                                          OUTPUT
      IF(JAER-2) 30,30,40                                          OUTPUT  30  40
   30 AP(49)= (T + LIFT*SINA-DB - DRAG*COSA) / W                   OUTPUT
      AP(50)= ( LIFT*COSA+ DRAG*SINA)/ W                           OUTPUT
      GO TO 50                                                     OUTPUT  50
   40 AP(49)= (T*COD +LIFT*SINA-DB -DRAG*COSA) / W                 OUTPUT
      AP(50)=(LIFT*COSA+DRAG*SINA-T*SID)/W                         OUTPUT
   50 AP(51)=SQRT(AP(49)**2+AP(50)**2)                             OUTPUT
      AP(27)= CL                                                   OUTPUT
      AP(28)= CD                                                   OUTPUT
      AP(37) = LV                                                  OUTPUT
      AP(38) = LGAM                                                OUTPUT
      AP(39) = LPSI                                                OUTPUT
      AP(40) = LR                                                  OUTPUT
      AP(41) = LRHO                                                OUTPUT
      AP(42) = LMU                                                 OUTPUT
      AP(44) = LM                                                  OUTPUT
      AP(47) = LTAU                                                OUTPUT
      CALL  COORDS(COSRHO,SINRHO)                                  OUTPUT
      IF(IPFLG1.NE.0) GO TO  60                                    OUTPUT  60
      AP(62)= SMIMAJ *FTNM                                         OUTPUT
      AP(63)=ECC                                                   OUTPUT
      AP(64)=AINCL * RAD                                           OUTPUT
      AP(65)= ASCNOD* RAD                                          OUTPUT
      AP(66)= ARGP * RAD                                           OUTPUT
      AP(67)= APOGEE * FTNM                                        OUTPUT
      AP(68)= PERGEE * FTNM                                        OUTPUT
      AP(69)= ANOMLY *RAD                                          OUTPUT
      AP(70)= SMIMAJ *2.*PI*SQRT(      SMIMAJ/GM)                  OUTPUT
```

```
      AP(71)=ENERGY                                                OUTPUT
      AP(72)=HMNTM                                                 OUTPUT
      AP(73) = P*FTNM                                              OUTPUT
      AP(74)=HMNTM /APOGEE                                         OUTPUT
      AP(75)=HMNTM / PERGEE                                        OUTPUT
   60 CONTINUE                                                     OUTPUT
      IF(IPFLG3. NE. 0) GO TO 70                                   OUTPUT
      IKONVR = 0                                                   OUTPUT
      IF(KONVER) IKONVR = 1                                        OUTPUT
      IF(.NOT.KONVER) GO TO 90                                     OUTPUT
      IF(V.LT.300..OR.ABS(COSGAM).LT..001) GO TO 70                JUL21
      CALL CRASH                                                   OUTPUT
   70 CONTINUE                                                     OUTPUT
      IF(.NOT.KONVER) GO TO 90                                     OUTPUT
      IF(IPFLG1.NE.0) GO  TO 90                                    OUTPUT
      AP(17) = Z (19)                                              OUTPUT
      AP(55) = Z(20)                                               OUTPUT
      AP(56) = Z(22)                                               OUTPUT
      AP(57) = Z(23)                                               OUTPUT
      AP(58) = Z(21)                                               OUTPUT
   90 CONTINUE                                                     OUTPUT
      CALL PRINT(ITER, IKONVR, 1, NOPRNT)                          OUTPUT
      NOPRNT = .FALSE.                                             OUTPUT
      RETURN                                                       OUTPUT
      END                                                          OUTPUT
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AINCL	i	I	Orbital inclination	(RAD)	/ORBIT /(	7)	OUTPUT	I	AINCL
							POBCQL	M	AINCL
ALPHA	α	I	Angle of attack	(RAD)	/DYNA /(	79)	AEROCO	I	ALPHA
							ALGCON	M	ALPHA
							AL2	I	ALPHA
							ARCIN	M	ALPHA
							CONTRL	M	ALPHA
							ENVPRO	I	ALPHA
							MOMECO	I	ALPHA
							NPLANE	M	ALPHA
							OUTPUT	I	ALPHA
							TRAJIN	O	ALPHA
							UT	I	ALPHA
							WRAPUP	I	ALPHA
ALT	h	I	Altitude	(FT)	/D /(	94)	OUTPUT	I	ALT
							STATEF	I	ALT
							WRAPUP	I	ALT
ANOMLY	ξ	I	True anomaly	(RAD)	/ORBIT /(	13)	OUTPUT	I	ANOMLY
							POBCQL	M	ANOMLY
APOGEE	R_a	I	Apogee radius	(FT)	/ORBIT /(	11)	OUTPUT	I	APOGEE
							POBCQL	O	APOGEE
ARGP	ω_p	I	Orbital argument of perigee	(RAD)	/ORBIT /(	8)	OUTPUT	I	ARGP
							POBCQL	M	ARGP
ASCNOD	Ω	I	Longitude of ascending node	(RAD)	/ORBIT /(	9)	OUTPUT	I	ASCNOD
							POBCQL	M	ASCNOD
CD	C_D	I	Drag coefficient		/DYNA /(	192)	AEROCO	O	CD
							OUTPUT	I	CD
							UT	I	CD
CL	C_L	I	Lift coefficient		/DYNA /(	186)	AEROCO	M	CL
							OUTPUT	I	CL
							UT	I	CL
COD	$\cos\delta_E$	I	See symbol		/DYNA /(	153)	DL2	I	COD
							OUTPUT	I	COD
							TH3	I	COD
							UT	M	COD
COSA	$\cos\alpha$	I	See symbol		/DYNA /(	10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
COSGAM	$\cos\gamma$	I	See symbol		/DYNA /(	4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							POBCQL	I	COSGAM
							STATEF	M	COSGAM
COSPHI	$\cos\phi$	I	See symbol		/DYNA /(	93)	AL1	I	COSPHI
							AL4	I	COSPHI
							APPLY	I	COSPHI
							ARCIN	O	COSPHI
							CONTRL	M	COSPHI
							OUTPUT	I	COSPHI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE US… SUBR	CODE	
COSRHO	$\cos\rho$	I	See symbol	/DYNA /(	97)	AL4	I	COS
						AL7	I	COS
						AL8	I	COS
						AL9	I	COS
						CONTRL	I	COS
						NLDRV	I	COS
						OUTPUT	I	COS
						POBCOL	I	COS
						STATEF	M	COS
CS	a	I	Speed of sound (FT/SEC)	/DYNA /(	16)	OUTPUT	I	CS
						STATEF	I	CS
DB	D_b	I	Base drag (LBS)	/DYNA /(	163)	AL1	I	DB
						AL4	I	DB
						AL6	I	DB
						AL7	I	DB
						AL8	I	DB
						AL9	I	DB
						APPLY	I	DB
						CONTRL	I	DB
						NLDRV	I	DB
						OUTPUT	I	DB
						STATEF	I	DB
						TH3	I	DB
						UT	I	DB
DELTAE	δ_E	I	Engine deflection (RADS)	/DYNA /(	155)	ALGCON	M	DEL
						ARCIN	M	DEL
						CONTRL	M	DEL
						DL1	I	DEL
						OUTPUT	I	DEL
						TRAJIN	O	DEL
						UT	I	DEL
DRAG	D	I	Aerodynamic drag (LBS)	/DYNA /(	69)	AL5	I	DRA
						AL7	I	DRA
						AL8	I	DRA
						AL9	I	DRA
						APPLY	I	DRA
						CONTRL	I	DRA
						ENVPRO	I	DRA
						NLDRV	I	DRA
						OUTPUT	I	DRA
						TH3	I	DRA
						UT	M	DRA
ECC	e	I	Orbital eccentricity	/ORBIT /(	6)	OUTPUT	I	ECC
						POBCOL	M	ECC
ENERGY	E	I	Energy	/ORBIT /(	17)	OUTPUT	I	ENE
						POBCOL	O	ENE
GAM	γ	I	Relative flight path angle. (RAD)	/D /(	92)	ARCIN	I	GAM
						ENVPRO	I	GAM
						OUTPUT	I	GAM
						STATEF	I	GAM
						WRAPUP	I	GAM
GAMI	γ_I	I	Inertial flight path angles (RAD)	/ORBIT /(	2)	OUTPUT	I	GAM
						POBCOL	O	GAM
GM	GM	I	Product of Newton's universal gravitational constant and the mass of the earth. (FT^3/SEC^2)	/GLOBAL/(	67)	OUTPUT	I	GM
						POBCOL	I	GM
						STATEF	I	GM
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						POBCOL	I	GR
						OLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
HMNTM	H	I	Momentum	/ORBIT /(	18)	OUTPUT PDBCOL	I M	HMNTM HMNTM
HT	Q	I	Heating (BTU)	/D /(	99)	OUTPUT WRAPUP	I I	HT HT
IATM		I	Atmosphere option flag	/ARCDAT/(	7)	ARCIN NLDRV OUTPUT STATEF	I I I I	IATM IATM IATM IATM
IPFLG1		I	IPFLG1≠0 supresses print-out of velocity losses and inertial Euler angles.	/GLOBAL/(	69)	OUTPUT PDBCOL QLTOSZ SALVE	I I O I	IPFLG1 IPFLG1 IPFLG1 IPFLG1
IPFLG3		I	IPFLG3≠0 supresses print-out of impact data.	/GLOBAL/(	71)	OUTPUT QLTOSZ	I O	IPFLG3 IPFLG3
ITER		I	QL iteration number.	/CNTRL /(	2)	ETIME GROPE OUTPUT	M M I	ITER ITER ITER
JAER		I	Aerodynamic model option flag	/ARCDAT/(	9)	AEROCO ARCIN OUTPUT STATEF UT	I I I I I	JAER JAER JAER JAER JAER
J3		I	Angle of attack option flag. J3 = 1: Optimal angle of attack; J3 = 2: Constant angle of attack; J3 = 3: Untrimmed lift limit; J3 = 4: Vertical rise or pitchover; J3 = 5: Unpowered total acceleration limit; J3 = 6: Gravity turn; J3 = 7: Dynamic pressure constraint; J3 = 8: Heating rate constraint; J3 = 9: Reynolds number constraint.	/DYNA /(	175)	ARCIN CONTRL NPLANE OUTPUT	O M M I	J3 J3 J3 J3
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL /(	28)	ALGCON APPLY ARCIN COHOMO GROPE NLDRV OUTPUT RKUTT1	I I I O O I I I	KONVER KONVER KONVER KONVER KONVER KONVER KONVER KONVER
LGAM	λ_γ	I	Relative flight path angle costate	/D /(	101)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LGAM LGAM LGAM LGAM LGAM LGAM
LIFT	L	I	Aerodynamic lift (LBS)	/DYNA /(	60)	AL4 AL5 AL6 APPLY CONTRL ENVPRO OUTPUT TH3 UT	I I I I I I I I O	LIFT LIFT LIFT LIFT LIFT LIFT LIFT LIFT LIFT
LM	λ_m	I	Mass costate	/D /(	106)	NLDRV OUTPUT WRAPUP	I I I	LM LM LM
LMU	λ_μ	I	Relative longitude costate	/D /(	105)	NLDRV OUTPUT WRAPUP	I I I	LMU LMU LMU
LPSI	λ_ψ	I	Relative azimuth angle costate	/D /(	102)	AL1 ARCIN CONTRL NLDRV OUTPUT WRAPUP	I I I I I I	LPSI LPSI LPSI LPSI LPSI LPSI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LR	λ_R	I	Altitude costate		/D	/(103)	NLDRV	I	LR
							OUTPUT	I	LR
							WRAPUP	I	LR
LRHO	λ_ρ	I	Latitude costate		/D	/(104)	NLDRV	I	LRHO
							OUTPUT	I	LRHO
							WRAPUP	I	LRHO
LTAU	λ_τ	I	Subarc duration costate		/D	/(107)	OUTPUT	I	LTAU
							WRAPUP	I	LTAU
LV	λ_v	I	Relative velocity costate		/D	/(100)	AL1	I	LV
							CONTRL	I	LV
							NLDRV	I	LV
							NPLANE	I	LV
							OUTPUT	I	LV
							WRAPUP	I	LV
M	m	I	Mass	(G'S)	/D	/(97)	AL4	I	M
							AL7	I	M
							AL8	I	M
							AL9	I	M
							APPLY	I	M
							BRANPT	I	M
							COSTAB	I	M
							COSTAI	I	M
							INTRPT	I	M
							NLDRV	I	M
							OUTPUT	I	M
							SALVE	I	M
							STATEF	I	M
							WRAPUP	I	M
MACH	M	I	Mach number		/DYNA	/(26)	AEROCO	I	MACH
							ENVPRQ	I	MACH
							OUTPUT	I	MACH
							STATEF	M	MACH
MU	μ	I	Relative longitude	(RAD)	/D	/(96)	OUTPUT	I	MU
							PDBCQL	I	MU
							WRAPUP	I	MU
NOPRNT		M	Not used.		/CNTRL	/(29)	OUTPUT	M	NOPRNT
							SALVE	O	NOPRNT
							TRAJIN	O	IPRNT
P	P_r	I	Semi-latus rectum	(FT)	/ORBIT	/(5)	OUTPUT	I	P
							PDBCQL	M	P
PA	P_a	I	Atmospheric pressure	(LBS/FT^2)	/DYNA	/(14)	IMPULS	I	PA
							NLDRV	I	PA
							OUTPUT	I	PA
							TH2	I	PA
PERGEE	R_p	I	Perigee radius	(FT)	/ORBIT	/(12)	OUTPUT	I	PERGEE
							PDBCQL	O	PERGEE
PHI	ϕ	I	Bank angle	(RAD)	/DYNA	/(80)	CONTRL	M	PHI
							OUTPUT	I	PHI
							WRAPUP	I	PHI
PSI	ψ	I	Relative azimuth angle.	(RAD)	/D	/(93)	OUTPUT	I	PSI
							STATEF	I	PSI
							WRAPUP	I	PSI
PSII	ψ_I	I	Inertial azimuth	(RAD)	/ORBIT	/(3)	OUTPUT	I	PSII
							PDBCQL	M	PSII
Q	q	I	Dynamic pressure	(LBS/FT^2)	/DYNA	/(27)	ENVPRQ	I	Q
							OUTPUT	I	Q
							PDBCQL	I	Q
							STATEF	M	Q
							UT	I	Q
RHO	ρ	I	Latitude	(RAD)	/D	/(95)	AL9	I	RHO
							OUTPUT	I	RHO
							STATEF	I	RHO
							WRAPUP	I	RHO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
RO	ρ_a	I	Atmospheric density (SLGS/FT^3)	/DYNA	/(15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCQL	I	RO
						STATEF	I	RO
SCROSS	S_C	I	Cross range (FT)	/ORBIT	/(149)	OUTPUT	I	SCROSS
						PDBCQL	O	SCROSS
SDOWN	S_D	I	Down range (FT)	/ORBIT	/(148)	OUTPUT	I	SDOWN
						PDBCQL	O	SDOWN
SID	$\sin\delta_E$	I	See symbol	/DYNA	/(154)	DL2	I	SID
						OUTPUT	I	SID
						TH3	I	SID
						UT	M	SID
SINA	$\sin\alpha$	I	See symbol	/DYNA	/(9)	AL1	I	SINA
						AL4	I	SINA
						AL6	I	SINA
						AL7	I	SINA
						AL8	I	SINA
						AL9	I	SINA
						APPLY	I	SINA
						CONTRL	I	SINA
						OUTPUT	I	SINA
						TH3	I	SINA
						UT	M	SINA
SINPHI	$\sin\phi$	I	See symbol	/DYNA	/(92)	AL1	I	SINPHI
						AL4	I	SINPHI
						APPLY	I	SINPHI
						CONTRL	M	SINPHI
						OUTPUT	I	SINPHI
SINRHO	$\sin\rho$	I	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCQL	I	SINRHO
						STATEF	M	SINRHO
SMIMAJ	a_s	I	Semi-major axis (FT)	/ORBIT	/(10)	OUTPUT	I	SMIMAJ
						PDBCQL	M	SMIMAJ
STOT	S_T	I	Total range (FT)	/ORBIT	/(158)	OUTPUT	I	STOT
						PDBCQL	O	STOT
						QLTOSZ	I	STOT
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
TAU	τ	I	Subarc duration (SEC)	/D	/(98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU

1164

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TIME		I	Trajectory time (SEC)	/DYNA	/(2)	ENVPRO	I	TIME
						OUTPUT	I	TIME
						PDBCQL	I	TIME
						STATEF	M	TIME
						WRAPUP	I	TIME
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCQL	I	V
						STATEF	I	V
						WRAPUP	I	V
VI	V_I	I	Inertial velocity (FT/SEC)	/ORBIT	/(1)	OUTPUT	I	VI
						PDBCQL	I	ORBPRM
						PDBCQL	M	VI
VNU	μ_a	I	Atmospheric viscosity (dynamic) (SLGS/FT/SEC)	/DYNA	/(214)	OUTPUT	I	VNU
						PDBCQL	I	VNU
W	W	I	Weight (LBS)	/DYNA	/(91)	AL5	I	W
						ENVPRO	I	W
						OUTPUT	I	W
						PDBCQL	I	W
						QLTOSZ	I	W
						STATEF	M	W
						TH3	I	W
XJ	$\dot{j}$	I	Control blend factor	/DYNA	/(159)	ARCIN	O	XJ
						DL2	I	XJ
						OUTPUT	I	XJ
						STATEF	I	XJ
						UT	I	XJ
XMCG	M_{CG}	I	Aerodynamic moment about center of gravity (FT-LBS)	/DYNA	/(160)	DL2	I	XMCG
						OUTPUT	I	XMCG
						UT	M	XMCG
XMUI	μ_I	I	Inertial longitude (RAD)	/ORBIT	/(4)	OUTPUT	I	XMUI
						PDBCQL	M	XMUI
XX		I	Fraction of subarc that has transpired	/DYNA	/(1)	ARCIN	O	XX
						ERROR	I	XX
						OUTPUT	I	XX
						STATEF	M	XX
Z	Z	I	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRO	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ZD		I	A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.1 of this document.	/ZD	/(1)	ENVPRO	I	ZD
						LINDRV	I	ZD
						OUTPUT	I	ZD
						RKUTT2	I	ZD
						WRAPUP	I	ZD

SUBRØUTINE PDBCQL

1156

1197

Purpose

PDBCQL computes all functional target conditions and their partials with respect to the state.*

*See documentation for Subroutine PDBC in the Steepest Descent module

```
      SUBROUTINE PDBCQL(KK, F, S, SD, IOP, ISKP)                         PDBCQL
      LOGICAL SWITCH, ILOAD                                              DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,        DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,      DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                                DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                  DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                         DYNA
      COMMON /DYNA/                                                      DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,    JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,    DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,    DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,    DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,    DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,    DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,    DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,    DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,    DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,    DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,    DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,    DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,    DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,    DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,    DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,    DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,    DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,    DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED     DYNA
      COMMON /DYNA/                                                      DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,    DYNA
     *ILOAD ,PKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,    DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,    DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,    DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,    DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,    JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                  AUG09
      COMMON/GLOBAL/                                                     GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,    GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20),  GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,        GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                           GLOBAL
      COMMON/ARCDAT/                                                     ARCDAT
     *SREF       ,EJ         ,XISP       ,TMULT      ,DTNC       ,DTPI       ,  ARCDAT
     *IATM       ,IMODE      ,JAER       ,JPRO       ,QMAX       ,GMAX       ,  ARCDAT
     *XLMAX      ,HDMAX      ,GMDOT      ,ALFMAX     ,PHMAX      ,MAEA       ,  ARCDAT
     *MAEB       ,MAEC       ,MAED       ,MAEE       ,MAEF       ,MAEG       ,  ARCDAT
     *MT         ,MISP       ,MXCG       ,MZCG       ,MWDA       ,MWDB       ,  ARCDAT
     *MDB        ,XCGR       ,ZCGR       ,XE         ,ZE         ,XT         ,  ARCDAT
     *DREF       ,MCND       ,RHOB       ,QMULT      ,REMAX      ,FRATE         ARCDAT
      DIMENSION  ARCDA(40)                                               ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                            ARCDAT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM    D
     *, LHT                                                              D
      COMMON /D/                                                         D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,   D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,    JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)   D
      DIMENSION NOM(20)                                                  D
      EQUIVALENCE (NOM, V)                                               D
C                                                                        PDBCQL
C        THIS  SUBROUTINE  COMPUTES  FUNCTIONS OF STATE  AND PARTIALS    PDBCQL
C        FOR  VARIOUS  PURPOSES  IN  TABTOP                              PDBCQL
C              ** DEFINITIONS**                                          PDBCQL
C                                                                        PDBCQL
C        KK  =  VARIABLE  CODE  NUMBER                                   PDBCQL
C        F   =  VALUE OF FUNCTION                                        PDBCQL
C        S   =  PARTIAL DERIVATIVE  ARRAY                                PDBCQL
C        SD  =  TIME DERIVATIVE OF F                                     PDBCQL
C        IOP =  OPTION FLAG = 0 =  COMPUTE F  ONLY                       PDBCQL
C                             1 =  COMPUTE S                             PDBCQL
C                             2 =  ESTIMATE SD  IF POSSIBLE              PDBCQL
C                             3 =  COMPUTE  SD                           PDBCQL
C        ISKP=  BYPASS FLAG  TO  INHIBIT  REDUNDANT  COMPUTATION         PDBCQL
C                                                                        PDBCQL
      DIMENSION S(1),F(1)                                                PDBCQL
```

```
      COMMON/ ORBIT/  VI,       GAMI,     PSII,     XMUI,         P,       ORBIT
     *     ECC,     AINCL,     ARGP,   ASCNOD,   SMIMAJ,   APOGEE,       ORBIT
     *  PERGEE,   ANOMLY,     CAPX,     CAPY,    ASYMP,   ENERGY,        ORBIT
     *   HMNTM,    DVIDV,    DVIDG,                                      ORBIT
     *  DVIDH ,    DVIDM,   DVIDPS,   DVIDRO,   DVIDMU,    DGIDV,        ORBIT
     *  DGIDG ,    DGIDH,   DGIDM ,   DGIDPS,   DGIDRO,    DGIDMU,       ORBIT
     *  DPIDV ,    DPIDG,   DPIDH ,   DPIDM ,   DPIDPS,    DPIDRO,       ORBIT
     *  DPIDMU,    DMIDV,   DMIDG ,   DMIDH ,   DMIDM ,    DMIDPS,       ORBIT
     *  DMIDRO,    DMIDMU,  DPDV  ,   DPDG  ,   DPDH  ,    DPDM  ,       ORBIT
     *  DPDPS ,    DPDRO ,  DPDMU ,   DECDV ,   DECDG ,    DECDH ,       ORBIT
     *  DECDM ,    DECDPS,  DECDRO,   DECDMU,   DIDV  ,    DIDG  ,       ORBIT
     *  DIDH  ,    DIDM  , DIDPS  ,  DIDRO  ,  DIDMU  ,   DBEDV  ,       ORBIT
     *  DBEDG ,    DBEDH ,  DBEDM ,   DBEDPS,   DBEDRO,    DBEDMU,       ORBIT
     *  DNODV ,    DNODG ,  DNODH ,   DNODM ,   DNODPS,    DNODRO,       ORBIT
     *  DNODMU,    DSMDV,    DSMDG,    DSMDH,    DSMDM,   DSMDPS         ORBIT
      COMMON/ORBIT/                                                      ORBIT
     *  DSMDRO,   DSMDMU,    DAPDV,    DAPDG,    DAPDH,   DAPDM ,        ORBIT
     *  DAPDPS,   DAPDRO,    DAPDMU,   DPEDV,    DPEDG,   DPEDH ,        ORBIT
     *DPEDM     ,DPEDPS     ,DPEDRO    ,DPEDMU    ,DANDV     ,DANDG    , ORBIT
     *  DANDH ,   DANDM ,    DANDPS,   DANDRO,   DANDMU,  DCXDV ,        ORBIT
     *  DCXDG ,   DCXDH ,    DCXDM ,   DCXDPS,   DCXDRO,  DCXDMU,        ORBIT
     *  DCYDV ,   DCYDG ,    DCYDH ,   DCYDM ,   DCYDPS,  DCYDRO,        ORBIT
     *  DCYDMU,   DASDV ,    DASDG ,   DASDH ,   DASDM ,  DASDPS,        ORBIT
     *  DASDRO,   DASDMU,    DENDV ,   DENDG ,   DENDH ,  DENDM ,        ORBIT
     *  DENDPS,   DENDRO,    DENDMU,   DMODV ,   DMODG ,  DMODH ,        ORBIT
     *  DMODM ,   DMODPS,    DMODRO,   DMODMU                            ORBIT
      DIMENSION ORBPRM(18),PPO(7,18)                                     ORBIT
      EQUIVALENCE (VI,ORBPRM),(DVIDV,PPO)                                ORBIT
      COMMON/ORBIT/ YAXRF,SNXLMR,CSXLMR,SDOWN,SCROSS,TD,TC               ORBIT
     *,SNPSR     ,CSPSR      ,SNGI       ,CSGI       ,SPSII      ,CPSII  , ORBIT
     *,STOT      ,CSI        ,SNI        ,SNGNU    - ,CSANO      ,COSOMU , ORBIT
     * SINOMU    ,THT        ,WTFUEL                                   , ORBIT
      DIMENSION JVAR(36),MASK(36), MS(2)                                 PDBCOL
      DIMENSION MAP(7)                                                   PDBCOL
      EQUIVALENCE   (MASK(1),MSK1),(MASK(2),MSK2),(MASK(3),MSK3),        PDBCOL
     1(MASK(4),MSK4), (MASK(5),MSK5)  ,(MASK(6),MSK6) ,(MASK(7),MSK7),   PDBCOL
     2(MASK(8),MSK8), (MASK(9),MSK9)  ,(MASK(10),MSK10),(MASK(11),MSK11) PDBCOL
     3,(MASK(12),MSK12),(MASK(13),MSK13),(MASK(14),MSK14),               PDBCOL
     4 (MASK(15),MSK15),(MASK(16),MSK16),(MASK(17),MSK17),               PDBCOL
     5 (MASK(18),MSK18),(MASK(19),MSK19),(MASK(20),MSK20),               PDBCOL
     6 (MASK(21),MSK21),(MASK(22),MSK22),(MASK(23),MSK23),               PDBCOL
     7 (MASK(24),MSK24),(MASK(25),MSK25),(MASK(26),MSK26),               PDBCOL
     8 (MASK(27),MSK27),(MASK(28),MSK28),(MASK(29),MSK29),               PDBCOL
     9 (MASK(30),MSK30),(MASK(31),MSK31),(MASK(32),MSK32),               PDBCOL
     X (MASK(33),MSK33),(MASK(34),MSK34),(MASK(35),MSK35),               PDBCOL
     A (MASK(36),MSK36)                                                  PDBCOL
      EQUIVALENCE (VNU, DYN214), (VNR, DYN215)                           PDBCOL
      DATA JVAR /0000001,0000003, 0000007,                               PDBCOL
     1 0000010,0000023, 0000043,                                         PDBCOL
     2 0000107,   0000367, 0000517, 0001001,                             PDBCOL
     3 0003043,  0005043,  0010023, 0020063,                             PDBCOL
     4 0040063, 0100043, 0201001, 0400003,                               PDBCOL
     501000001     ,03000003       ,04000007       ,010000000    ,       PDBCOL
     6023000023    ,043000043      ,0104000107     ,0367000367   ,       PDBCOL
     70504000517   ,01001001001    ,03043001043    ,05043001043  ,       PDBCOL
     8010023010023 ,020036000063   ,040036000063   ,010004310043 ,       PDBCOL
     9  01001001001,  01003400003  /                                     PDBCOL
      DATA  MASK / 1,2, 4, 8, 16, 32 , 64, 128, 256 , 512 , 1024, 2048,  PDBCOL
     14096, 8192,16384, 32768 , 65536 , 131072 , 262144 ,  524288,       PDBCOL
     2  1048576 ,    2097152 ,   4194304 , 8388608 , 16777216        ,   PDBCOL
     3   33554432 , 67108864,  134217728, 268435456,   536870912  ,      PDBCOL
     4 1073741824 ,  2147483648 ,4294967296,8589934592 ,                 PDBCOL
     5 17179869184,0/                                                    PDBCOL
      DATA MS/ 03000003, 07000007/                                       PDBCOL
      DATA MAP /1, 2, 4, 7, 3, 5, 6/                                     PDBCOL
      DO 1 ILOOP = 1, 9                                                  PDBCOL
    1 S(ILOOP) = 0.                                                      PDBCOL
      OCORHO = OMEGA*COSRHO                                              PDBCOL
      IF(KK.GT.0) GO TO  10                                              PDBCOL
      K=1                                                                PDBCOL
      MM = 0617777                                                       PDBCOL
      GO TO 50                                                           PDBCOL
   10 CONTINUE                                                           PDBCOL
      IF(KK.GE.30) GO TO 530                                             PDBCOL
C           PARAMETER IS  AMONG ORBITAL SET                              PDBCOL
```

6 OCT 72 6.01-44

```
C       TEST  FOR  ESTIMATE  OPTION                                      PDBCQL
        K=KK-11                                                          PDBCQL
        IF(IOP.EQ.0) GO TO 20                                            PDBCQL
C           TEST  FOR  PARAMETER CALCULATION ONLY                        PDBCQL
        K=18+K                                                           PDBCQL
     20 IF(ISKP.EQ.0) GO TO 40                                           PDBCQL
     30 MM=AND(JVAR(K),-MS(ISKP))                                        PDBCQL
        IF(MM.NE.0)  GO TO 50                                            PDBCQL
        IF(K-18)230,230,420                                              PDBCQL
     40 MM = JVAR(K)                                                     PDBCQL
C         TEST  AND  COMPUTE  VI                                         PDBCQL
     50 IF(0.EQ.AND(MM,MSK1))  GO  TO  60                                PDBCQL
        VE = R*OMEGA*COSRHO                                              PDBCQL
        VI=  SQRT( V*V +2.*VE*V*COSGAM*SINPSI+VE*VE)                     PDBCQL
        IF(MM.LT.MSK2) GO TO 230                                         PDBCQL
C         TEST  AND  COMPUTE  GAMMA  INERTIAL  AND ITS TRIG FUNCTIONS    PDBCQL
     60 IF( AND(MM,MSK2).EQ.0) GO TO 70                                  PDBCQL
        SNGI = V*SINGAM/VI                                               PDBCQL
        GAMI = ASIN(SNGI)                                                PDBCQL
        CSGI = SQRT(1.- SNGI*SNGI)                                       PDBCQL
        ISKP =1                                                          PDBCQL
        IF(MM.LT.MSK3)GO TO 230                                          PDBCQL
C        TEST AND  COMPUTE   PSI INERTIAL  AND  ITS  TRIG  FUNCTIONS     PDBCQL
     70 IF( AND(MM,MSK3).EQ.0) GO TO 80                                  PDBCQL
        SPSII=(V*COSGAM*SINPSI+VE)/(VI *CSGI)                            PDBCQL
        PSII = ATAN2(V*COSGAM*SINPSI+VE     ,V*COSGAM*COSPSI)            PDBCQL
        CPSII= COS(PSII)                                                 PDBCQL
        ISKP =2                                                          PDBCQL
        IF(MM.LT.MSK4) GO TO 230                                         PDBCQL
C         TEST AND COMPUTE   MU INERTIAL                                 PDBCQL
     80 IF( AND(MM,MSK4).EQ.0) GO TO 90                                  PDBCQL
        XMUI= MU+ OMGZ*TIME                                              PDBCQL
        IF(KK .LT. 0 .AND. IPFLG1 .NE. 0) GO TO 540                      PDBCQL
        IF(MM.LT.MSK5) GO  TO 230                                        PDBCQL
C         TEST  AND  COMPUTE   SEMI-LATUS  RECTUM                        PDBCQL
     90 IF(AND(MM,MSK5).EQ.0) GO  TO  100                                PDBCQL
        P= R*R* VI*VI* CSGI*CSGI /GM                                     PDBCQL
        IF(MM.LT.MSK6) GO TO 230                                         PDBCQL
C         TEST  AND  COMPUTE  ECCENTRICITY                               PDBCQL
    100 IF(AND(MM,MSK6).EQ.0) GO TO 110                                  PDBCQL
        RV2OMU =  R*VI*VI/GM                                             PDBCQL
        ECC=  SQRT( 1. - RV2OMU*(2.-RV2OMU)*CSGI*CSGI )                  PDBCQL
        IF(MM.LT.MSK7) GO TO 230                                         PDBCQL
C         TEST  AND  COMPUTE  INCLINATION  AND ITS  TRIG  FUNCTIONS      PDBCQL
    110 IF(AND(MM,MSK7).EQ.0) GO TO 120                                  PDBCQL
        CSI=  COSRHO* SPSII                                              PDBCQL
        AINCL = ACOS(CSI)                                                PDBCQL
        SNI =  SIN(AINCL)                                                PDBCQL
        IF(MM.LT.MSK8) GO TO 230                                         PDBCQL
C         TEST  AND  COMPUTE  ARG. OF  PERIGEE                           PDBCQL
    120 IF(AND(MM,MSK8).EQ.0) GO TO  130                                 PDBCQL
        CSGNU =(P -R)/R/ECC                                              PDBCQL
        IF(ABS(CSGNU) .GT. 1.) CSGNU = SIGN(1., CSGNU)                   PDBCQL
        GNU = ACOS(CSGNU)                                                PDBCQL
        SROI = SINRHO/SNI                                                PDBCQL
        IF(ABS(SROI).GT.1.)  SROI = SIGN(1.,SROI)                        PDBCQL
        ARGP = ASIN( SROI) - GNU                                         PDBCQL
        IF(MM.LT.MSK9) GO  TO 230                                        PDBCQL
C         TEST  AND  COMPUTE  LONG. OF  ASCENDING  NODE                  PDBCQL
    130 IF(AND(MM,MSK9).EQ.0) GO TO 140                                  PDBCQL
        SPSI = SPSII*SINRHO/SNI                                          PDBCQL
        IF(ABS(SPSI).GT.1.)  SPSI=  SIGN(1.,SPSI)                        PDBCQL
        ASCNOD = XMUI - ASIN(SPSI)                                       PDBCQL
        IF(MM.LT.MSK10) GO TO 230                                        PDBCQL
C         TEST  AND  COMPUTE  SEMI-MAJOR AXIS                            PDBCQL
    140 IF(AND(MM,MSK10).EQ.0) GO TO 150                                 PDBCQL
        SMIMAJ=   R*GM /(2.*GM -R*VI*VI)                                 PDBCQL
        IF(MM.LT.MSK11) GO TO 230                                        PDBCQL
C         TEST  AND  COMPUTE  APOGEE                                     PDBCQL
```

```
  150 IF(AND(MM,MSK11).EQ.0) GO TO 160                              PDBCQL
      APOGEE=  SMIMAJ*(1.+ECC)                                      PDBCQL
      IF(MM LT.MSK12) GO TO 230                                     PDBCQL
C        TEST  AND  COMPUTE  PERIGEE                                PDBCQL
  160 IF(AND(MM,MSK12).EQ.0)  GO TO 170                             PDBCQL
      PERGEE =  SMIMAJ*(1.-ECC)                                     PDBCQL
      IF(MM.LT.MSK13) GO TO 230                                     PDBCQL
C        TEST  AND  COMPUTE  TRUE  ANOMOLY                          PDBCQL
  170 IF(AND(MM,MSK13).EQ.0) GO TO 180                              PDBCQL
      ANOMLY =  ATAN2( SNGI/CSGI, 1.- R/ P )                        PDBCQL
      IF(MM.LT.MSK14) GO TO 230                                     PDBCQL
C     TEST  AND  COMPUTE   CAP X                                    PDBCQL
  180 IF(AND(MM,MSK14).EQ.0) GO TO 190                              PDBCQL
      CAPX =  P/ECC                                                 PDBCQL
      IF(MM.LT.MSK15) GO TO 230                                     PDBCQL
C       TEST  AND   COMPUTE  CAPY                                   PDBCQL
  190 IF(AND(MM,MSK15).EQ.0) GO TO 200                              PDBCQL
      CAPY =  P*ECC / SQRT( ECC*ECC -1)                             PDBCQL
      IF(MM.LT.MSK16) GO TO 230                                     PDBCQL
C       TEST  AND  COMPUTE  ASSYMPTOTE                              PDBCQL
  200 IF(AND(MM,MSK16).EQ.0) GO TO 210                              PDBCQL
      ASYMP = ACOS(1./ECC)                                          PDBCQL
      IF(MM.LT.MSK17) GO   TO 230                                   PDBCQL
C       TEST  AND  COMPUTE  ENERGY                                  PDBCQL
  210 IF(AND(MM,MSK17).EQ.0) GO TO  220                             PDBCQL
      ENERGY =    2.*GM / SMIMAJ                                    PDBCQL
      IF(MM.LT.MSK18) GO TO 230                                     PDBCQL
C       TEST  AND  COMPUTE  MOMENTUM                                PDBCQL
  220 IF(AND(MM,MSK18).EQ.0) GO TO  240                             PDBCQL
       HMNTM=  R*VI*CSGI                                            PDBCQL
      IF(MM.GE.MSK19) GO TO 240                                     PDBCQL
      IF(KK.LT.0) GO TO 540                                         PDBCQL
C              RETURN  REQ. ORB. PARA. THROUGH ARG. LIST            PDBCQL
  230 F =  ORBPRM(K)                                                PDBCQL
      RETURN                                                        PDBCQL
C        TEST AND COMPUTE  PARTIALS  OF  V  INERTIAL                PDBCQL
  240 IF(AND(MM,MSK19).EQ.0) GO TO 250                              PDBCQL
      DVIDV =(V+R*DCORHO*COSGAM*SINPSI)/VI                          PDBCQL
      SNGI=V*SINGAM/VI                                              PDBCQL
      DVIDG =            -VE*SNGI*SINPSI                            PDBCQL
      DVIDH =            VE*(V * COSGAM* SINPSI+ R*DCORHO)/ (R*VI)  PDBCQL
      DVIDM =0.                                                     PDBCQL
      DVIDPS=            VE*V*COSGAM*COSPSI/ VI                     PDBCQL
      DVIDRO= -SINRHO*VE*(V*COSGAM*SINPSI+ VE) / (COSRHO*VI)        PDBCQL
      DVIDMU= 0.                                                    PDBCQL
      IF(MM.LT.MSK20) GO TO 420                                     PDBCQL
C       TEST AND COMPUTE  PARTIALS  OF   GAMMA  INERTIAL            PDBCQL
  250 IF(AND(MM,MSK20).EQ.0) GO TO 260                              PDBCQL
      VICI = VI* CSGI                                               PDBCQL
      TMQ= - V* SINGAM/(VI* VICI)                                   PDBCQL
      DGIDV = SINGAM/VICI + TMQ*DVIDV                               PDBCQL
      DGIDG = V*COSGAM/VICI + TMQ*DVIDG                             PDBCQL
      DGIDH = TMQ*DVIDH                                             PDBCQL
      DGIDM=0.                                                      PDBCQL
      DGIDPS= TMQ* DVIDPS                                           PDBCQL
      DGIDRO= TMQ* DVIDRO                                           PDBCQL
      DGIDMU=0.                                                     PDBCQL
      IF(MM.LT.MSK21) GO TO 420                                     PDBCQL
C       TEST AND  COMPUTE  PARTIALS  PSI  INERTIAL                  PDBCQL
  260 IF(AND(MM,MSK21).EQ.0) GO TO 270                              PDBCQL
      TM1= CPSII*CPSII /(V*COSGAM*COSPSI)                           PDBCQL
      DPIDV=-VE* TM1/ V                                             PDBCQL
      DPIDH= -DPIDV*V/R                                             PDBCQL
      DPIDG= DPIDH*R*SINGAM/COSGAM                                  PDBCQL
      DPIDPS=                                                       PDBCQL
     1     CPSII*CPSII/( COSPSI*COSPSI)+SINPSI*VE *TM1 /COSPSI      PDBCQL
      DPIDRO= -R*DMGZ*SINRHO*TM1                                    PDBCQL
      DPIDMU= 0.                                                    PDBCQL
      DPIDM = 0.                                                    PDBCQL
```

```
      IF(MM.LT.MSK22) GO TO 420                                         PDBCOL
C          TEST  AND  COMPUTE  PARTIALS  OF  MU  INERTIAL               PDBCOL
  270 IF(AND(MM,MSK22).EQ.0) GO TO 280                                  PDBCOL
      DMIDV =0.                                                         PDBCOL
      DMIDG =0.                                                         PDBCOL
      DMIDH =0.                                                         PDBCOL
      DMIDM =0.                                                         PDBCOL
      DMIDPS=0.                                                         PDBCOL
      DMIDRO=0.                                                         PDBCOL
      DMIDMU=1.                                                         PDBCOL
      IF(MM.LT.MSK23) GO TO 420                                         PDBCOL
C        TEST AND  COMPUTE PARTIALS  OF  SEMI-LATUS RECTUM              PDBCOL
  280 IF(AND(MM,MSK23).EQ.0) GO TO 290                                  PDBCOL
      TM2= 2.* P /VI                                                    PDBCOL
      TM3= 2.* P / R                                                    PDBCOL
      TM4=-2.*P/CSGI *SNGI                                              PDBCOL
      DPDV= TM2*DVIDV + TM4*DGIDV                                       PDBCOL
      DPDG= TM2*DVIDG + TM4*DGIDG                                       PDBCOL
      DPDH= TM3 + TM2*DVIDH + TM4*DGIDH                                 PDBCOL
      DPDM= 0.                                                          PDBCOL
      DPDPS= TM2*DVIDPS + TM4*DGIDPS                                    PDBCOL
      DPDRO= TM2*DVIDRO + TM4*DGIDRO                                    PDBCOL
      DPDMU= 0.                                                         PDBCOL
      IF(MM.LT.MSK24) GO TO 420                                         PDBCOL
C      TEST  AND  COMPUTE  PARTIALS  OF  ECCENTRICITY                   PDBCOL
  290 IF(AND(MM,MSK24).EQ.0) GO TO 300                                  PDBCOL
      TMM=  (RV2OMU -1 )*RV2OMU*CSGI*CSGI/ECC                           PDBCOL
      TM5= 2.*TMM/VI                                                    PDBCOL
      TM6 =  RV2OMU*(2. - RV2OMU) * CSGI* SNGI/ ECC                     PDBCOL
      TM7 = TMM/ R                                                      PDBCOL
      DECDV= TM5*DVIDV + TM6*DGIDV                                      PDBCOL
      DECDG= TM5*DVIDG + TM6*DGIDG                                      PDBCOL
      DECDH= TM7+ TM5*DVIDH + TM6*DGIDH                                 PDBCOL
      DECDM=0.                                                          PDBCOL
      DECDPS=TM5*DVIDPS + TM6*DGIDPS                                    PDBCOL
      DECDRO=TM5*DVIDRO + TM6*DGIDRO                                    PDBCOL
      DECDMU= 0.                                                        PDBCOL
      IF(MM.LT.MSK25) GO TO 420                                         PDBCOL
C        TEST AND COMPUTE PARTIALS  OF  INCLINATION                     PDBCOL
  300 IF(AND(MM,MSK25).EQ.0) GO TO 310                                  PDBCOL
      TM8 = -COSRHO* CPSII /SNI                                         PDBCOL
      DIDV=  TM8*DPIDV                                                  PDBCOL
      DIDG=  TM8*DPIDG                                                  PDBCOL
      DIDH=  TM8*DPIDH                                                  PDBCOL
      DIDM=  0.                                                         PDBCOL
      DIDPS= TM8*DPIDPS                                                 PDBCOL
      DIDRO= SINRHO*SPSII /SNI + TM8*DPIDRO                             PDBCOL
      DIDMU= 0.                                                         PDBCOL
      IF(MM.LT.MSK26) GO TO 420                                         PDBCOL
C      TEST  AND  COMPUTE  PARTIALS  OF  ARG. OF PERIGEE                PDBCOL
  310 IF(AND(MM,MSK26).EQ.0) GO TO  320                                 PDBCOL
      CGNAR  =  COS(GNU +ARGP)                                          PDBCOL
      SNGNU = SIN(GNU)                                                  PDBCOL
      DGNDP=  -1./(R*ECC*SNGNU)                                         PDBCOL
      DGNDR =  -P*DGNDP/ R                                              PDBCOL
      DGNDE =  -(P-R)/ECC *DGNDP                                        PDBCOL
      TM9 = - SINRHO*CSI/SNI/SNI/CGNAR                                  PDBCOL
      DBEDV =-DGNDP*DPDV - DGNDE*DECDV + TM9*DIDV                       PDBCOL
      DBEDG =-DGNDP*DPDG - DGNDE*DECDG + TM9*DIDG                       PDBCOL
      DBEDH = -DGNDR -DGNDP*DPDH -DGNDE*DECDH + TM9*DIDH                PDBCOL
      DBEDM =0.                                                         PDBCOL
      DBEDRO= COSRHO/(SNI*CGNAR)-DGNDP*DPDRO -DGNDE*DECDRO +TM9*DIDRO   PDBCOL
      DBEDPS= -DGNDP*DPDPS -DGNDE*DECDPS +TM9*DIDPS                     PDBCOL
      DBEDMU= 0.                                                        PDBCOL
      IF(MM.LT.MSK27) GO TO 420                                         PDBCOL
C          TEST AND COMPUTE  PARTIALS OF  LONG.OF ASCENDING NODE        PDBCOL
  320 IF(AND(MM,MSK27).EQ.0) GO TO 330                                  PDBCOL
      CDNO =-COS(ASCNOD -MU- OMGZ*TIME)                                 PDBCOL
      DNODPI = CPSII *SINRHO/(SNI*CDNO)                                 PDBCOL
      DNODRR = SPSII*COSRHO/(SNI*CDNO)                                  PDBCOL
      DNODI = - SPSII*SINRHO*CSI /(SNI*SNI*CDNO)                        PDBCOL
      DNODV = DNODPI*DPIDV +DNODI*DIDV                                  PDBCOL
```

```
      DNODG = DNODPI*DPIDG +DNODI*DIDG                              PDBCQL
      DNODH = DNODPI*DPIDH +DNODI*DIDH                              PDBCQL
      DNODM =0.                                                     PDBCQL
      DNODPS=DNODPI*DPIDPS +DNODI*DIDPS                             PDBCQL
      DNODRO= DNODRR +DNODPI*DPIDRO + DNODI*DIDRO                   PDBCQL
      DNODMU= 1.                                                    PDBCQL
      IF(MM.LT.MSK28) GO TO 420                                     PDBCQL
C     TEST AND COMPUTE  PARTIALS OF  SMIMAJ                         PDBCQL
  330 IF(AND(MM,MSK28).EQ.0) GO TO 340                              PDBCQL
C     TEST  FOR  MOMENTUM                                           PDBCQL
      IF(K.EQ.36) GO TO 410                                         PDBCQL
      DSMDR =  SMIMAJ/R*(1. + SMIMAJ*VI*VI/GM)                      PDBCQL
      DSMDVI=  2.*SMIMAJ*SMIMAJ*VI /GM                              PDBCQL
      DSMDV = DSMDVI*DVIDV                                          PDBCQL
      DSMDG = DSMDVI*DVIDG                                          PDBCQL
      DSMDH= DSMDR + DSMDVI*DVIDH                                   PDBCQL
      DSMDM = 0.                                                    PDBCQL
      DSMDPS= DSMDVI* DVIDPS                                        PDBCQL
      DSMDRO= DSMDVI*DVIDRO                                         PDBCQL
      DSMDMU = 0.                                                   PDBCQL
C      TEST  FOR  ENERGY                                            PDBCQL
      IF(K.EQ.35) GO TO 400                                         PDBCQL
      IF(MM.LT.MSK29) GO TO 420                                     PDBCQL
C      TEST AND  COMPUTE PARTIALS OF APOGEE                         PDBCQL
  340 IF(AND(MM,MSK29).EQ.0) GO TO 350                              PDBCQL
      DAPDA = 1.+ECC                                                PDBCQL
      DAPDE =  SMIMAJ                                               PDBCQL
      DAPDV= DAPDA*DSMDV + DAPDE*DECDV                              PDBCQL
      DAPDG =DAPDA*DSMDG + DAPDE*DECDG                              PDBCQL
      DAPDH=DAPDA*DSMDH +DAPDE*DECDH                                PDRCQL
      DAPDM = 0.                                                    PDBCQL
      DAPDPS= DAPDA*DSMDPS + DAPDE*DECDPS                           PDBCQL
      DAPDRO= DAPDA*DSMDRO + DAPDE*DECDRO                           PDBCQL
      DAPDMU= 0.                                                    PDBCQL
      IF(MM.LT.MSK30) GO TO 420                                     PDBCQL
C       TEST AND COMPUTE  PARTIALS OF  PERIGEE                      PDBCQL
  350 IF(AND(MM,MSK30).EQ.0) GO TO  360                             PDBCQL
      DPEDA =  1.-ECC                                               PDBCQL
      DPEDE =  -SMIMAJ                                              PDBCQL
      DPEDV =  DPEDA*DSMDV + DPEDE*DECDV                            PDBCQL
      DPEDG =  DPEDA*DSMDG + DPEDE*DECDG                            PDBCQL
      DPEDH =  DPEDA*DSMDH + DPEDE*DECDH                            PDBCQL
      DPEDM = 0                                                     PDBCQL
      DPEDPS=  DPEDA*DSMDPS +DPEDE*DECDPS                           PDBCQL
      DPEDRO=  DPEDA*DSMDRO +DPEDE*DECDRO                           PDBCQL
      DPEDMU= 0.                                                    PDBCQL
      IF(MM.LT MSK31) GO TO 420                                     PDBCQL
C       TEST  AND  COMPUTE PARTIALS OF  ANOMOLY                     PDBCQL
  360 IF(AND(MM,MSK31).EQ.0) GO TO 370                              PDBCQL
      CSANO =  COS(ANOMLY)                                          PDBCQL
      CS2  =  CSANO*CSANO                                           PDBCQL
      CON  =  1. - R/P                                              PDBCQL
      DANDGI = CS2/ CSGI/ CSGI/ CON                                 PDBCQL
      DANDR  =  CS2*SNGI/(P*CON*CON*CSGI)                           PDBCQL
      DANDP  =  - DANDR * R/P                                       PDBCQL
      DANDV = DANDGI*DGIDV + DANDP*DPDV                             PDBCQL
      DANDG = DANDGI*DGIDG + DANDP*DPDG                             PDBCQL
      DANDH = DANDR + DANDGI*DGIDH + DANDP*DPDH                     PDBCQL
      DANDM =0.                                                     PDBCQL
      DANDPS = DANDGI*DGIDPS + DANDP*DPDPS                          PDBCQL
      DANDRO = DANDGI*DGIDRO + DANDP*DPDRO                          PDBCQL
      DANDMU = 0.                                                   PDBCQL
      IF(MM.LT.MSK32) GO TO 420                                     PDBCQL
C         TEST   AND   COMPUTE  PARTIALS  OF  CAPX                  PDBCQL
  370 IF(AND(MM,MSK32).EQ.0) GO TO  380                             PDBCQL
      DCXDP  = 1./ ECC                                              PDBCQL
      DCXDE  = -P/(ECC*ECC)                                         PDBCQL
      DCXDV = DCXDP*DPDV + DCXDE*DECDV                              PDBCQL
      DCXDG = DCXDP*DPDG + DCXDE*DECDG                              PDBCQL
      DCXDH = DCXDP*DPDH + DCXDE*DECDH                              PDBCQL
      DCXDM =0.                                                     PDBCQL
      DCXDPS= DCXDP*DPDPS +DCXDE*DECDPS                             PDBCQL
      DCXDRO= DCXDP*DPDRO +DCXDE*DECDRO                             PDBCQL
```

```
          DCXDMU= 0.                                                     PDBCOL
          IF(MM.LT.MSK33) GO TO 420                                      PDBCOL
C             TEST   AND   COMPUTE  PARTIALS  OF   CAPY                  PDBCOL
      380 IF(AND(MM,MSK33).EQ.0)  GO  TO 390                             PDBCOL
          TM1   =        ECC*ECC                                         PDBCOL
          TM2   =  TM1-1                                                 PDBCOL
          SQE2  =  SQRT(TM2)                                             PDBCOL
          DCYDP = ECC/ SQE2                                              PDBCOL
          DCYDE =   P/SQE2*( 1. - TM1/ TM2)                              PDBCOL
          DCYDV =DCYDP*DPDV + DCYDE*DECDV                                PDBCOL
          DCYDG =DCYDP  *DPDG +DCYDE*DECDG                               PDBCOL
          DCYDH =DCYDP*DPDH + DCYDE*DECDH                                PDBCOL
          DCYDM =0                                                       PDBCOL
          DCYDPS =DCYDP*DPDPS +DCYDE* DECDPS                             PDBCOL
          DCYDRO = DCYDP*DPDRO +DCYDE* DECDRO                            PDBCOL
          DCYDMU = 0.                                                    PDBCOL
          IF(MM.LT.MSK34) GO TO 420                                      PDBCOL
C             TEST   AND   COMPUTE  PARTIALS  OF   ASSYMPTOTE            PDBCOL
      390 IF(AND(MM,MSK34).EQ. 0)  GO  TO 420                            PDBCOL
          SNASY = SIN(ASYMP)                                             PDBCOL
          DASDE = 1./ ( SNASY*ECC*ECC)                                   PDBCOL
          DASDV =  DASDE*DECDV                                           PDBCOL
          DASDG =  DASDE*DECDG                                           PDBCOL
          DASDH =  DASDE*DECDH                                           PDBCOL
          DASDM = 0.                                                     PDBCOL
          DASDPS=  DASDE*DECDPS                                          PDBCOL
          DASDRO = DASDE* DECDRO                                         PDBCOL
          DASDMU =  0.                                                   PDBCOL
          GO TO  420                                                     PDBCOL
C                COMPUTE   PARTIALS OF- ENERGY                           PDBCOL
      400 CONTINUE                                                       PDBCOL
          DENDSM= -2.*GM / (SMIMAJ* SMIMAJ)                              PDBCOL
          DENDV = DENDSM * DSMDV                                         PDBCOL
          DENDG = DENDSM * DSMDG                                         PDBCOL
          DENDH = DENDSM * DSMDH                                         PDBCOL
          DENDM = 0.                                                     PDBCOL
          DENDPS= DENDSM * DSMDPS                                        PDBCOL
          DENDRO= DENDSM * DSMDRO                                        PDBCOL
          DENDMU= 0.                                                     PDBCOL
          GO   TO  420                                                   PDBCOL
C           COMPUTE  PARTIALS  OF  MOMENTUM                              PDBCOL
      410 CONTINUE                                                       PDBCOL
          DMODR=HMNTM/R                                                  PDBCOL
          DMODVI=  HMNTM / VI                                            PDBCOL
          DMODGI= - R*VI * SNGI                                          PDBCOL
          DMODV =  DMODVI*DVIDV + DMODGI *DGIDV                          PDBCOL
          DMODG =  DMODVI*DVIDG+DMODGI*DGIDG                             PDBCOL
          DMODH =  DMODR  + DMODVI*DVIDH +DMODGI*DGIDH                   PDBCOL
          DMODM =0.                                                      PDBCOL
          DMODPS =DMODVI*DVIDPS +           DMODGI*DGIDPS                PDBCOL
          DMODRO =DMODVI*DVIDRO +           DMODGI*DGIDRO                PDBCOL
          DMODMU =0.                                                     PDBCOL
C             STORE  PARTIALS IN  S  AND COMPUTE  SD                     PDBCOL
      420 K=K-18                                                         PDBCOL
          DO 430 J=1,7                                                   PDBCOL
          JMAP = MAP(J)                                                  PDBCOL
          S(JMAP) = PPD(J, K)                                            PDBCOL
      430 CONTINUE                                                       PDBCOL
          F = ORBPRM(K)                                                  PDBCOL
          RETURN                                                         PDBCOL
C           KK.GE.30   REENTRY TYPE TARGETS                              PDBCOL
      530 K= KK-29                                                       PDBCOL
          GO TO (540,570,600,630,690,750,800)                            PDBCOL
         * ,K                                                            PDBCOL
C            DOWNRANGE  CALCULATION                                      PDBCOL
      540 COSDMU = COS(MU-YMXRF)                                         PDBCOL
          SINDMU = SIN(MU-YMXRF)                                         PDBCOL
          TDNUM= CSPSR*(SINRHO*CSXLMR-COSRHO*SNXLMR*COSDMU)+SNPSR*COSRHO*SIN PDBCOL
         1DMU                                                            PDBCOL
          TDDEN= SINRHO*SNXLMR +COSRHO*CSXLMR*COSDMU                     PDBCOL
          TD = ATAN2(TDNUM,TDDEN)                                        PDBCOL
          SDOWN =TD*ER                                                   PDBCOL
```

```
      IF(KK.LT.0) GO TO 580                                              PDBCOL  580
      F= TD*ER                                                           PDBCOL
C           PARTIALS  OF  DOWNRANGE                                      PDBCOL
  550 COSTD = COS(TD)                                                    PDBCOL
      S(5)  =ER*COSTD*COSTD*((CSPSR*(COSRHO*CSXLMR+SINRHO*SNXLMR*COSDMU) PDBCOL
     1   -SNPSR*SINRHO*SINDMU)/TDDEN - TDNUM*(COSRHO*SNXLMR -SINRHO*CSXL PDBCOL
     2MR *  COSDMU)/(TDDEN*TDDEN) )                                      PDBCOL
      S(6)  =ER*COSTD*COSTD*((  CSPSR*COSRHO*SNXLMR*SINDMU +SNPSR*COSRHO PDBCOL
     1*COSDMU)/TDDEN  + TDNUM*COSRHO*CSXLMR*SINDMU/(TDDEN*TDDEN) )       PDBCOL
  560 RETURN                                                             PDBCOL
C          CROSS  RANGE  COMPUTATION                                     PDBCOL
  570 COSDMU = COS(MU-YMXRF)                                             PDBCOL
      SINDMU = SIN(MU-YMXRF)                                             PDBCOL
  580 CONTINUE                                                           PDBCOL
      STC = CSPSR *COSRHO*SINDMU - SNPSR*(SINRHO*CSXLMR -COSRHO*SNXLMR*  PDBCOL
     1         COSDMU)                                                   PDBCOL
      TC = ASIN(STC)                                                     PDBCOL
      SCROSS=TC*ER                                                       PDBCOL
      IF(KK.LT.0) GO TO 610                                              PDBCOL  610
      F=TC*ER                                                            PDBCOL
C          PARTIALS   OF    CROSS RANGE                                  PDBCOL
  590 COSTC  COS(TC)                                                     PDBCOL
      S(5)= ER/COSTC*(-CSPSR*SINRHO*SINDMU - SNPSR*(COSRHO*CSXLMR+SINRHO PDBCOL
     1*   SNXLMR* COSDMU))                                               PDBCOL
      S(6) = ER/COSTC*(CSPSR*COSRHO*COSDMU - SNPSR*COSRHO*SNXLMR*SINDMU) PDBCOL
      RETURN                                                             PDBCOL
C        TOTAL RANGE                                                     PDBCOL
  600 COSDMU = COS(MU-YMXRF)                                             PDBCOL
  610 CONTINUE                                                           PDBCOL
      COSTHT = SINRHO* SNXLMR  + COSRHO*CSXLMR * COSDMU                  PDBCOL
      THT  =  ACOS(COSTHT)                                               PDBCOL
      STOT = ER*THT                                                      PDBCOL
      IF(KK.LT.0) RETURN                                                 PDBCOL
      F= ER*THT                                                          PDBCOL
  620 SNTHT . SIN(THT)                                                   PDBCOL
      S(5)= -ER*(COSRHO*SNXLMR -SINRHO*CSXLMR*COSDMU) / SNTHT            PDBCOL
      SINDMU= SIN(MU-YMXRF)                                              PDBCOL
      S(6) = COSRHO*CSXLMR*SINDMU/SNTHT*ER                               PDBCOL
      GO TO  560                                                         PDBCOL  560
C          DYNAMIC  PRESSURE                                             PDBCOL
  630 CONTINUE                                                           PDBCOL
      F= Q                                                               PDBCOL
C           PARTIALS  OF  DYNAMIC PRESSURE                               PDBCOL
  640 S(1) = QV                                                          PDBCOL
      S(4) = QR                                                          PDBCOL
      RETURN                                                             PDBCOL
C           HEATING  RATE                                                PDBCOL
  690 CONTINUE                                                           PDBCOL
      HTD = QMULT*17600.*SQRT(RO/RHOB)*(V/26000.)**3.15                  PDBCOL
      F = HTD                                                            PDBCOL
C            HEATING  RATE  PARTIALS                                     PDBCOL
  700 S(1) = 3.15*HTD/V                                                  PDBCOL
      S(4) = HTD/2.*ROR/RO                                               PDBCOL
      RETURN                                                             PDBCOL
C        REYNOLDS NUMBER                                                 PDBCOL
  750 RE = V*RO/VNU                                                      PDBCOL
      F = RE                                                             PDBCOL
C           PARTIALS  OF  REYNOLS  NUMBER                                PDBCOL
      S(1) = RO/VNU                                                      PDBCOL
      S(4) = (V*ROR - RE*VNR)/VNU                                        PDBCOL
      RETURN                                                             PDBCOL
C        FUEL  WEIGHT   (FIRST STAGE ONLY)                               PDBCOL
  800 WTFUEL = ZSAVE(7)*GR - W                                           PDBCOL
      F= WTFUEL                                                          PDBCOL
C        FUEL WT  PARTIALS                                               PDBCOL
      S(7) = -GR                                                         PDBCOL
      RETURN                                                             PDBCOL
      END                                                                PDBCOL
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AINCL	i	M	Orbital inclination	(RAD)	/ORBIT /(	7)	OUTPUT	I	AINCL
							PDBCQL	M	AINCL
ANOMLY	ζ	M	True anomoly	(RAD)	/ORBIT /(	13)	OUTPUT	I	ANOMLY
							PDBCQL	M	ANOMLY
APOGEE	R_a	O	Apogee radius	(FT)	/ORBIT /(	11)	OUTPUT	I	APOGEE
							PDBCQL	O	APOGEE
ARGP	ω_p	M	Orbital argument of perigee	(RAD)	/ORBIT /(	8)	OUTPUT	I	ARGP
							PDBCQL	M	ARGP
ASCNOD	Ω	M	Longitude of ascending node	(RAD)	/ORBIT /(	9)	OUTPUT	I	ASCNOD
							PDBCQL	M	ASCNOD
ASYMP	Θ	M	Outgoing asymptote	(RAD)	/ORBIT /(	16)	PDBCQL	M	ASYMP
CAPX	X	O	Asymptote parameter	T	/ORBIT /(	14)	PDBCQL	O	CAPX
CAPY	Y	O	Asymptote parameter	(FT)	/ORBIT /(	15)	PDBCQL	O	CAPY
COSDMU	$\cos(\mu-\mu_r)$	M	See symbol		/ORBIT /(	163)	PDBCQL	M	COSDMU
COSGAM	$\cos\gamma$	I	See symbol		/DYNA /(	4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCQL	I	COSGAM
							STATEF	M	COSGAM
COSPSI	$\cos\psi$	I	See symbol		/DYNA /(	95)	AL4	I	COSPSI
							AL7	I	COSPSI
							AL8	I	COSPSI
							AL9	I	COSPSI
							CONTRL	I	COSPSI
							NLDRV	I	COSPSI
							PDBCQL	I	COSPSI
							STATEF	O	COSPSI
COSRHO	$\cos\rho$	I	See symbol		/DYNA /(	97)	AL4	I	COSRHO
							AL7	I	COSRHO
							AL8	I	COSRHO
							AL9	I	COSRHO
							CONTRL	I	COSRHO
							NLDRV	I	COSRHO
							OUTPUT	I	COSRHO
							PDBCQL	I	COSRHO
							STATEF	M	COSRHO
CPSII	$\cos(\psi_I)$	M	Cosine of inertial azimuth		/ORBIT /(	157)	PDBCQL	M	CPSII
CSANO	$\cos(\zeta)$	M	See symbol		/ORBIT /(	162)	PDBCQL	M	CSANO
CSGI	$\cos(\gamma_I)$	M	Cosine of inertial flight path angle		/ORBIT /(	155)	PDBCQL	M	CSGI
CSI	$\cos(i)$	M	Cosine of inclination		/ORBIT /(	159)	PDBCQL	M	CSI
CSPSR	$\cos(\psi_r)$	I	Cosine of reference azimuth		/ORBIT /(	153)	CHECK	O	CSPSR
							PDBCQL	I	CSPSR
CSXLMR	$\cos(\rho-\rho_r)$	I	Cosine of refernece latitude		/ORBIT /(	147)	CHECK	O	CSXLMR
							PDBCQL	I	CSXLMR
DANDG		O	Partial derivative of boundary condition		/ORBIT /(	104)	PDBCQL	O	DANDG
DANDH		O	Partial derivative of boundary condition		/ORBIT /(	105)	PDBCQL	O	DANDH
DANDM		O	Partial derivative of boundary condition		/ORBIT /(	106)	PDBCQL	O	DANDM
DANDMU		O	Partial derivative of boundary condition		/ORBIT /(	109)	PDBCQL	O	DANDMU
DANDPS		O	Partial derivative of boundary condition		/ORBIT /(	107)	PDBCQL	O	DANDPS
DANDRO		O	Partial derivative of boundary condition		/ORBIT /(	108)	PDBCQL	O	DANDRO
DANDV		O	Partial derivative of boundary condition		/ORBIT /(	103)	PDBCQL	O	DANDV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DAPDG		0	Partial derivative of boundary condition	/ORBIT /(	90)	PDBCQL	0	DAPDG
DAPDH		0	Partial derivative of boundary condition	/ORBIT /(	91)	PDBCQL	0	DAPDH
DAPDM		0	Partial derivative of boundary condition	/ORBIT /(	92)	PDBCQL	0	DAPDM
DAPDMU		0	Partial derivative of boundary condition	/ORBIT /(	95)	PDBCQL	0	DAPDMU
DAPDPS		0	Partial derivative of boundary condition	/ORBIT /(	93)	PDBCQL	0	DAPDPS
DAPDRO		0	Partial derivative of boundary condition	/ORBIT /(	94)	PDBCQL	0	DAPDRO
DAPDV		0	Partial derivative of boundary condition	/ORBIT /(	89)	PDBCQL	0	DAPDV
DASDG		0	Partial derivative of boundary condition	/ORBIT /(	125)	PDBCQL	0	DASDG
DASDH		0	Partial derivative of boundary condition	/ORBIT /(	126)	PDBCQL	0	DASDH
DASDM		0	Partial derivative of boundary condition	/ORBIT /(	127)	PDBCQL	0	DASDM
DASDMU		0	Partial derivative of boundary condition	/ORBIT /(	130)	PDBCQL	0	DASDMU
DASDPS		0	Partial derivative of boundary condition	/ORBIT /(	128)	PDBCQL	0	DASDPS
DASDRO		0	Partial derivative of boundary condition	/ORBIT /(	129)	PDBCQL	0	DASDRO
DASDV		0	Partial derivative of boundary condition	/ORBIT /(	124)	PDBCQL	0	DASDV
DBEDG		0	Partial derivative of boundary condition	/ORBIT /(	69)	PDBCQL	0	DBEDG
DBEDH		0	Partial derivative of boundary condition	/ORBIT /(	70)	PDBCQL	0	DBEDH
DBEDM		0	Partial derivative of boundary condition	/ORBIT /(	71)	PDBCQL	0	DBEDM
DBEDMU		0	Partial derivative of boundary condition	/ORBIT /(	74)	PDBCQL	0	DBEDMU
DBEDPS		0	Partial derivative of boundary condition	/ORBIT /(	72)	PDBCQL	0	DBEDPS
DBEDRO		0	Partial derivative of boundary condition	/ORBIT /(	73)	PDBCQL	0	DBEDRO
DBEDV		0	Partial derivative of boundary condition	/ORBIT /(	68)	PDBCQL	0	DBEDV
DCXDG		0	Partial derivative of boundary condition	/ORBIT /(	111)	PDBCQL	0	DCXDG
DCXDH		0	Partial derivative of boundary condition	/ORBIT /(	112)	PDBCQL	0	DCXDH
DCXDM		0	Partial derivative of boundary condition	/ORBIT /(	113)	PDBCQL	0	DCXDM
DCXDMU		0	Partial derivative of boundary condition	/ORBIT /(	116)	PDBCQL	0	DCXDMU
DCXDPS		0	Partial derivative of boundary condition	/ORBIT /(	114)	PDBCQL	0	DCXDPS
DCXDRO		0	Partial derivative of boundary condition	/ORBIT /(	115)	PDBCQL	0	DCXDRO
DCXDV		0	Partial derivative of boundary condition	/ORBIT /(	110)	PDBCQL	0	DCXDV
DCYDG		0	Partial derivative of boundary condition	/ORBIT /(	118)	PDBCQL	0	DCYDG
DCYDH		0	Partial derivative of boundary condition	/ORBIT /(	119)	PDBCQL	0	DCYDH
DCYDM		0	Partial derivative of boundary condition	/ORBIT /(	120)	PDBCQL	0	DCYDM
DCYDMU		0	Partial derivative of boundary condition	/ORBIT /(	123)	PDBCQL	0	DCYDMU
DCYDPS		0	Partial derivative of boundary condition	/ORBIT /(	121)	PDBCQL	0	DCYDPS
DCYDRO		0	Partial derivative of boundary condition	/ORBIT /(	122)	PDBCQL	0	DCYDRO
DCYDV		0	Partial derivative of boundary condition	/ORBIT /(	117)	PDBCQL	0	DCYDV
DECDG		M	Partial derivative of boundary condition	/ORBIT /(	55)	PDBCQL	M	DECDG
DECDH		M	Partial derivative of boundary condition	/ORBIT /(	56)	PDBCQL	M	DECDH
DECDM		0	Partial derivative of boundary condition	/ORBIT /(	57)	PDBCQL	0	DECDM
DECDMU		0	Partial derivative of boundary condition	/ORBIT /(	60)	PDBCQL	0	DECDMU
DECDPS		M	Partial derivative of boundary condition	/ORBIT /(	58)	PDBCQL	M	DECDPS
DECDRO		M	Partial derivative of boundary condition	/ORBIT /(	59)	PDBCQL	M	DECDRO
DECDV		M	Partial derivative of boundary condition	/ORBIT /(	54)	PDBCQL	M	DECDV
DENDG		0	Partial derivative of boundary condition	/ORBIT /(	132)	PDBCQL	0	DENDG
DENDH		0	Partial derivative of boundary condition	/ORBIT /(	133)	PDBCQL	0	DENDH
DENDM		0	Partial derivative of boundary condition	/ORBIT /(	134)	PDBCQL	0	DENDM
DENDMU		0	Partial derivative of boundary condition	/ORBIT /(	137)	PDBCQL	0	DENDMU
DENDPS		0	Partial derivative of boundary condition	/ORBIT /(	135)	PDBCQL	0	DENDPS
DENDRO		0	Partial derivative of boundary condition	/ORBIT /(	136)	PDBCQL	0	DENDRO
DENDV		0	Partial derivative of boundary condition	/ORBIT /(	131)	PDBCQL	0	DENDV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DGIDG		M	Partial derivative of boundary condition	/ORBIT /(	27)	PDBCOL	M	DGIDG
DGIDH		M	Partial derivative of boundary condition	/ORBIT /(	28)	PDBCOL	M	DGIDH
DGIDM		O	Partial derivative of boundary condition	/ORBIT /(	29)	PDBCOL	O	DGIDM
DGIDMU		O	Partial derivative of boundary condition	/ORBIT /(	32)	PDBCOL	O	DGIDMU
DGIDPS		M	Partial derivative of boundary condition	/ORBIT /(	30)	PDBCOL	M	DGIDPS
DGIDRO		M	Partial derivative of boundary condition	/ORBIT /(	31)	PDBCOL	M	DGIDRO
DGIDV		M	Partial derivative of boundary condition	/ORBIT /(	26)	PDBCOL	M	DGIDV
DIDG		M	Partial derivative of boundary condition	/ORBIT /(	62)	PDBCOL	M	DIDG
DIDH		M	Partial derivative of boundary condition	/ORBIT /(	63)	PDBCOL	M	DIDH
DIDM		O	Partial derivative of boundary condition	/ORBIT /(	64)	PDBCOL	O	DIDM
DIDMU		O	Partial derivative of boundary condition	/ORBIT /(	67)	PDBCOL	O	DIDMU
DIDPS		M	Partial derivative of boundary condition	/ORBIT /(	65)	PDBCOL	M	DIDPS
DIDRO		M	Partial derivative of boundary condition	/ORBIT /(	66)	PDBCOL	M	DIDRO
DIDV		M	Partial derivative of boundary condition	/ORBIT /(	61)	PDBCOL	M	DIDV
DMIDG		O	Partial derivative of boundary condition	/ORBIT /(	41)	PDBCOL	O	DMIDG
DMIDH		O	Partial derivative of boundary condition	/ORBIT /(	42)	PDBCOL	O	DMIDH
DMIDM		O	Partial derivative of boundary condition	/ORBIT /(	43)	PDBCOL	O	DMIDM
DMIDMU		O	Partial derivative of boundary condition	/ORBIT /(	46)	PDBCOL	O	DMIDMU
DMIDPS		O	Partial derivative of boundary condition	/ORBIT /(	44)	PDBCOL	O	DMIDPS
DMIDRO		O	Partial derivative of boundary condition	/ORBIT /(	45)	PDBCOL	O	DMIDRO
DMIDV		O	Partial derivative of boundary condition	/ORBIT /(	40)	PDBCOL	O	DMIDV
DMODG		O	Partial derivative of boundary condition	/ORBIT /(	139)	PDBCOL	O	DMODG
DMODH		O	Partial derivative of boundary condition	/ORBIT /(	140)	PDBCOL	O	DMODH
DMODM		O	Partial derivative of boundary condition	/ORBIT /(	141)	PDBCOL	O	DMODM
DMODMU		O	Partial derivative of boundary condition	/ORBIT /(	144)	PDBCOL	O	DMODMU
DMODPS		O	Partial derivative of boundary condition	/ORBIT /(	142)	PDBCOL	O	DMODPS
DMODRO		O	Partial derivative of boundary condition	/ORBIT /(	143)	PDBCOL	O	DMODRO
DMODV		O	Partial derivative of boundary condition	/ORBIT /(	138)	PDBCOL	O	DMODV
DNODG		O	Partial derivative of boundary condition	/ORBIT /(	76)	PDBCOL	O	DNODG
DNODH		O	Partial derivative of boundary condition	/ORBIT /(	77)	PDBCOL	O	DNODH
DNODM		O	Partial derivative of boundary condition	/ORBIT /(	78)	PDBCOL	O	DNODM
DNODMU		O	Partial derivative of boundary condition	/ORBIT /(	81)	PDBCOL	O	DNODMU
DNODPS		O	Partial derivative of boundary condition	/ORBIT /(	79)	PDBCOL	O	DNODPS
DNODRO		O	Partial derivative of boundary condition	/ORBIT /(	80)	PDBCOL	O	DNODRO
DNODV		O	Partial derivative of boundary condition	/ORBIT /(	75)	PDBCOL	O	DNODV
DPDG		M	Partial derivative of boundary condition	/ORBIT /(	48)	PDBCOL	M	DPDG
DPDH		M	Partial derivative of boundary condition	/ORBIT /(	49)	PDBCOL	M	DPDH
DPDM		O	Partial derivative of boundary condition	/ORBIT /(	50)	PDBCOL	O	DPDM
DPDMU		O	Partial derivative of boundary condition	/ORBIT /(	53)	PDBCOL	O	DPDMU
DPDPS		M	Partial derivative of boundary condition	/ORBIT /(	51)	PDBCOL	M	DPDPS
DPDRO		M	Partial derivative of boundary condition	/ORBIT /(	52)	PDBCOL	M	DPDRO
DPDV		M	Partial derivative of boundary condition	/ORBIT /(	47)	PDBCOL	M	DPDV
DPEDG		O	Partial derivative of boundary condition	/ORBIT /(	97)	PDBCOL	O	DPEDG
DPEDH		O	Partial derivative of boundary condition	/ORBIT /(	98)	PDBCOL	O	DPEDH
DPEDM		O	Partial derivative of boundary condition	/ORBIT /(	99)	PDBCOL	O	DPEDM
DPEDMU		O	Partial derivative of boundary condition	/ORBIT /(	102)	PDBCOL	O	DPEDMU
DPEDPS		O	Partial derivative of boundary condition	/ORBIT /(	100)	PDBCOL	O	DPEDPS
DPEDRO		O	Partial derivative of boundary condition	/ORBIT /(	101)	PDBCOL	O	DPEDRO
DPEDV		O	Partial derivative of boundary condition	/ORBIT /(	96)	PDBCOL	O	DPEDV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DPIDG		M	Partial derivative of boundary condition	/ORBIT /(	34)	PDBCQL	M	DPIDG
DPIDH		M	Partial derivative of boundary condition	/ORBIT /(	35)	PDBCQL	M	DPIDH
DPIDM		O	Partial derivative of boundary condition	/ORBIT /(	36)	PDBCQL	O	DPIDM
DPIDMU		O	Partial derivative of boundary condition	/ORBIT /(	39)	PDBCQL	O	DPIDMU
DPIDPS		M	Partial derivative of boundary condition	/ORBIT /(	37)	PDBCQL	M	DPIDPS
DPIDRO		M	Partial derivative of boundary condition	/ORBIT /(	38)	PDBCQL	M	DPIDRO
DPIDV		M	Partial derivative of boundary condition	/ORBIT /(	33)	PDBCQL	M	DPIDV
DSMDG		M	Partial derivative of boundary condition	/ORBIT /(	83)	PDBCQL	M	DSMDG
DSMDH		M	Partial derivative of boundary condition	/ORBIT /(	84)	PDBCQL	M	DSMDH
DSMDM		O	Partial derivative of boundary condition	/ORBIT /(	85)	PDBCQL	O	DSMDM
DSMDMU		O	Partial derivative of boundary condition	/ORBIT /(	88)	PDBCQL	O	DSMDMU
DSMDPS		M	Partial derivative of boundary condition	/ORBIT /(	86)	PDBCQL	M	DSMDPS
DSMDRO		M	Partial derivative of boundary condition	/ORBIT /(	87)	PDBCQL	M	DSMDRO
DSMDV		M	Partial derivative of boundary condition	/ORBIT /(	82)	PDBCQL	M	DSMDV
DVIDG		M	Partial derivative of boundary condition	/ORBIT /(	20)	PDBCQL	M	DVIDG
DVIDH		M	Partial derivative of boundary condition	/ORBIT /(	21)	PDBCQL	M	DVIDH
DVIDM		O	Partial derivative of boundary condition	/ORBIT /(	22)	PDBCQL	O	DVIDM
DVIDMU		O	Partial derivative of boundary condition	/ORBIT /(	25)	PDBCQL	O	DVIDMU
DVIDPS		M	Partial derivative of boundary condition	/ORBIT /(	23)	PDBCQL	M	DVIDPS
DVIDRO		M	Partial derivative of boundary condition	/ORBIT /(	24)	PDBCQL	M	DVIDRO
DVIDV		M	Partial derivative of boundary condition	/ORBIT /(	19)	PDBCQL	M	DVIDV
						PDBCQL	I	PPD
ECC	e	M	Orbital eccentricity	/ORBIT /(	6)	OUTPUT	I	ECC
						PDBCQL	M	ECC
ENERGY	E	O	Energy	/ORBIT /(	17)	OUTPUT	I	ENERGY
						PDBCQL	O	ENERGY
ER	E_R	I	Earth radius. (FT)	/GLOBAL/(	2)	ENVPRO	I	ER
						PDBCQL	I	ER
						QLTOSZ	I	ER
						STATEF	I	ER
GAMI	γ_I	O	Inertial flight path angles (RAD)	/ORBIT /(	2)	OUTPUT	I	GAMI
						PDBCQL	O	GAMI
GM	GM	I	Product of Newton's universal gravitational constant and the mass of the earth. (FT^3/SEC^2)	/GLOBAL/(	67)	OUTPUT	I	GM
						PDBCQL	I	GM
						STATEF	I	GM
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCQL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
HMNTM	H	M	Momentum	/ORBIT /(	18)	OUTPUT	I	HMNTM
						PDBCQL	M	HMNTM
IPFLG1		I	IPFLG1≠0 supresses print-out of velocity losses and inertial Euler angles.	/GLOBAL/(	69)	OUTPUT	I	IPFLG1
						PDBCQL	I	IPFLG1
						QLTOSZ	O	IPFLG1
						SALVE	I	IPFLG1
MU	μ	I	Relative longitude (RAD)	/D /(	96)	OUTPUT	I	MU
						PDBCQL	I	MU
						WRAPUP	I	MU

10 NOV 72 G.01-47

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
OMEGA	ω	I	Earth rotation rate (RAD/SEC)	/DYNA /(	5)	AL4 AL7 CONTRL PDBCQL TRAJIN	I I I I M	OMEGA OMEGA OMEGA OMEGA OMEGA
OMGZ	ω	I	Earth rotation rate (RAD/SEC)	/GLOBAL/(	3)	PDBCQL TRAJIN	I I	OMGZ OMGZ
ORBPRM	V_I	I	Inertial velocity (FT/SEC)	/ORBIT /(	1)	OUTPUT PDBCQL PDBCQL	I I M	VI ORBPRM VI
P	P_r	M	Semi-latus rectum (FT)	/ORBIT /(	5)	OUTPUT PDBCQL	I M	P P
PERGEE	R_p	O	Perigee radius (FT)	/ORBIT /(	12)	OUTPUT PDBCQL	I O	PERGEE PERGEE
PPO		I	Partial derivative of boundary condition	/ORBIT /(	19)	PDBCQL PDBCQL	M I	DVIDV PPO
PSII	ψ_I	M	Inertial azimuth (RAD)	/ORBIT /(	3)	OUTPUT PDBCQL	I M	PSII PSII
Q	q	I	Dynamic pressure (LBS/FT^2)	/DYNA /(	27)	ENVPRO OUTPUT PDBCQL STATEF UT	I I I M I	Q Q Q Q Q
QMULT	=0 OR 1	I	Heating flag multiplier	/ARCDAT/(	40)	NLORV PDBCQL	I I	QMULT QMULT
QR	$\partial q/\partial R$	I	See symbol	/DYNA /(	29)	PDBCQL STATEF UT	I M I	QR QR QR
QV	$\partial q/\partial V$	I	See symbol	/DYNA /(	28)	PDBCQL STATEF UT	I M I	QV QV QV
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA /(	7)	AL4 AL7 AL8 AL9 CONTRL ENVPRO NLORV PDBCQL QLTOSZ STATEF	I I I I I I I I I M	R R R R R R R R R R
RHOB	ρ_b	I	Atmosphere base density for heating calculation (LB/FT**3)	/ARCDAT/(	39)	NLORV PDBCQL	I I	RHOB RHOB
RO	ρ_a	I	Atmospheric density (SLGS/FT^3)	/DYNA /(	15)	AL7 AL8 AL9 NLORV OUTPUT PDBCQL STATEF	I I I I I I I	RO RO RO RO RO RO RO
ROR	$\partial \rho_a/\partial R$	I	See symbol	/DYNA /(	19)	AL7 AL8 AL9 NLORV PDBCQL STATEF	I I I I I I	ROR ROR ROR ROR ROR ROR
SCROSS	S_c	O	Cross range (FT)	/ORBIT /(	149)	OUTPUT PDBCQL	I O	SCROSS SCROSS
SDOWN	S_D	O	Down range (FT)	/ORBIT /(	148)	OUTPUT PDBCQL	I O	SDOWN SDOWN
SINDMU	$\sin(\mu-\mu_r)$	M	See symbol	/ORBIT /(	164)	PDBCQL	M	SINDMU

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SINGAM	$\sin\gamma$	I	See symbol		/DYNA /(	3)	AL1	I	SINGAM
							AL4	I	SINGAM
							AL7	I	SINGAM
							AL8	I	SINGAM
							AL9	I	SINGAM
							CONTRL	I	SINGAM
							NLDRV	I	SINGAM
							PDBCQL	I	SINGAM
							STATEF	M	SINGAM
SINPSI	$\sin\psi$	I	See symbol		/DYNA /(	94)	AL4	I	SINPSI
							AL7	I	SINPSI
							AL8	I	SINPSI
							AL9	I	SINPSI
							CONTRL	I	SINPSI
							NLDRV	I	SINPSI
							PDBCQL	I	SINPSI
							STATEF	O	SINPSI
SINRHO	$\sin\rho$	I	See symbol		/DYNA /(	96)	AL4	I	SINRHO
							AL7	I	SINRHO
							AL8	I	SINRHO
							AL9	I	SINRHO
							CONTRL	I	SINRHO
							NLDRV	I	SINRHO
							OUTPUT	I	SINRHO
							PDBCQL	I	SINRHO
							STATEF	M	SINRHO
SMIMAJ	a_s	M	Semi-major axis	(FT)	/ORBIT /(	10)	OUTPUT	I	SMIMAJ
							PDBCQL	M	SMIMAJ
SNGI	$\sin(\gamma_I)$	M	Sine of inertial flight path angle		/ORBIT /(	154)	PDBCQL	M	SNGI
SNGNU	$\sin(\nu)$	M	See symbol		/ORBIT /(	161)	PDBCQL	M	SNGNU
SNI	$\sin(I)$	M	Sine of inclination		/ORBIT /(	160)	PDBCQL	M	SNI
SNPSR	$\sin(\psi_r)$	I	Sine of reference azimuth		/ORBIT /(	152)	CHECK	O	SNPSR
							PDBCQL	I	SNPSR
SNXLMR	$\sin(\rho-\rho_r)$	I	Sine of reference latitude		/ORBIT /(	146)	CHECK	O	SNXLMR
							PDBCQL	I	SNXLMR
SPSII	$\sin(\psi_I)$	M	Sin of inertial azimuth		/ORBIT /(	156)	PDBCQL	M	SPSII
STOT	S_T	O	Total range	(FT)	/ORBIT /(	158)	OUTPUT	I	STOT
							PDBCQL	O	STOT
							QLTOSZ	I	STOT
TC	θ_C	M	Cross range angle	(RAD)	/ORBIT /(	151)	PDBCQL	M	TC
TD	θ_D	M	Down range angle	(RAD)	/ORBIT /(	150)	PDBCQL	M	TD
THT	θ_T	M	Total range angle	(FT)	/ORBIT /(	165)	PDBCQL	M	THT
TIME		I	Trajectory time	(SEC)	/DYNA /(	2)	ENVPRQ	I	TIME
							OUTPUT	I	TIME
							PDBCQL	I	TIME
							STATEF	M	TIME
							WRAPUP	I	TIME

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
V	V	I	Relative velocity. (FT/SEC)	/D	/(91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRQ	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCQL	I	V
						STATEF	I	V
						WRAPUP	I	V
VI	V_I	M	Inertial velocity (FT/SEC)	/ORBIT	/(1)	OUTPUT	I	VI
						PDBCQL	I	ORBPRM
						PDBCQL	M	VI
VMR	$\partial\mu_a/\partial R$	I	See symbol	/DYNA	/(215)	PDBCQL	I	VMR
VMU	μ_a	I	Atmospheric viscosity (dynamic) (SLGS/FT/SEC)	/DYNA	/(214)	OUTPUT	I	VMU
						PDBCQL	I	VMU
W	W	I	Weight (LBS)	/DYNA	/(91)	AL5	I	W
						ENVPRQ	I	W
						OUTPUT	I	W
						PDBCQL	I	W
						QLTOSZ	I	W
						STATEF	M	W
						TN3	I	W
XMUI	μ_I	M	Inertial longitude (RAD)	/ORBIT	/(9)	OUTPUT	I	XMUI
						PDBCQL	M	XMUI
YMXRF	ρ_r	I	Reference longitude (RAD)	/ORBIT	/(145)	CHECK	O	YMXRF
						PDBCQL	I	YMXRF
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D	/(151)	BCOND	O	ZSAVE
						BRANPT	I	ZSAVE
						COSTAB	I	ZSAVE
						COSTAI	I	ZSAVE
						INTRPT	I	ZSAVE
						PDBCQL	I	ZSAVE
						SALVE	I	ZSAVE

SUBRØUTINE
QLTØSZ

Purpose

QLTOSZ handles the interface between the QL trajectory module and the Phase I or II (SSSP) sizing modules.*

*See documentation of subroutine TRTOSZ in Steepest Descent Module

QLTOSZ

```
      SUBROUTINE QLTOSZ                                                  JUL21
      COMMON /Y/ Y(820)                                                  Y
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM    D
     *, LHT                                                              D
      COMMON /D/                                                         D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,   D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,    JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)   D
      DIMENSION NOM(20)                                                  D
      EQUIVALENCE (NOM, V)                                               D
      COMMON /CNTRL/                                                     CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,    CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,    CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,    CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES           CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE-                    CNTRL
      LOGICAL SWITCH, ILOAD                                              DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,       DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,      DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                                DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                  DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                         DYNA
      COMMON /DYNA/                                                      DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,    JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,    DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,    DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,    DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,    DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,    DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,    DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,    DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,    DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,    DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,    DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,    DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,    DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,    DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,    DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,    DYNA
     *COOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,    DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,    DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED     DYNA
      COMMON /DYNA/                                                      DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,    DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,    DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,    DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,    DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,    DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,    JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                  AUG09
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C           PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),      QP(14),    EROR,      PZ(5),     VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,      TLNG,                    SIZING
C         PHASE I SIZING PARAMERERS                                      SIZING
     *WBO,      WLOO,      DWEB,      DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,     NNB,       WEO,       WEB,       WO,       WLD,    SIZING
     *DVO,      DVB,       MUB,       MUO,       VSTG,      WPO              SIZING
     *,JTYP,    BECO,      BSTG,      ORBI,      ITNBW,     ITNOW,    ISZD(23) SIZING
C            SAVES   TRAJECTORY-TO-SIZING DATA  DURING SOLUTION TRAJECTOR JUL21
C            ENTRY  POINTS  INCLUDE                                      JUL21
C            INTERI   SAVE  INIT. STATES                                 JUL21
C            ARCEND   SAVES INTERMEDIATE  DATA                           JUL21
C            TRJEND   SAVES  FINAL WTS. AND IMPULSIVE VELOCITY           JUL21
C                                                                        JUL21
      COMMON /PRINT/ AP(100)                                             JUL21
      COMMON/ ORBIT/  VI,       GAMI,      PSII,      XMUI,        P,    ORBIT
     *     ECC,       AINCL,      ARGP,   ASCNOD,    SMIMAJ,   APOGEE,   ORBIT
     *  PERGEE,    ANOMLY,      CAPX,      CAPY,     ASYMP,   ENERGY,   ORBIT
     *    HMNTM,     DVIDV,     DVIDG,                                   ORBIT
```

```
     *  DVIDH ,     DVIDM,    DVIDPS,    DVIDRO,    DVIDMU,    DGIDV,        ORBIT
     *  DGIDG ,     DGIDH,    DGIDM ,    DGIDPS,    DGIDRO,    DGIDMU,       ORBIT
     *  DPIDV ,     DPIDG,    DPIDH ,    DPIDM ,    DPIDPS,    DPIDRO,       ORBIT
     *  DPIDMU,     DMIDV,    DMIDG ,    DMIDH ,    DMIDM ,    DMIDPS,       ORBIT
     *  DMIDRO,     DMIDMU,   DPDV  ,    DPDG  ,    DPDH  ,    DPDM  ,       ORBIT
     *  DPDPS ,     DPDRO ,   DPDMU ,    DECDV ,    DECDG ,    DECDH ,       ORBIT
     *  DECDM ,     DECDPS,   DECDRO,    DECDMU,    DIDV  ,    DIDG  ,       ORBIT
     *  DIDH  ,     DIDM  ,  DIDPS  ,   DIDRO  ,   DIDMU  ,   DBEDV  ,       ORBIT
     *  DBEDG ,     DBEDH ,   DBEDM ,    DBEDPS,    DBEDRO,    DBEDMU,       ORBIT
     *  DNODV ,     DNODG ,   DNODH ,    DNODM ,    DNODPS,    DNODRO,       ORBIT
     *  DNODMU,     DSMDV,     DSMDG,     DSMDH,     DSMDM,    DSMDPS        ORBIT
      COMMON/ORBIT/                                                          ORBIT
     *  DSMDRO,    DSMDMU,    DAPDV,     DAPDG,     DAPDH,    DAPDM ,        ORBIT
     *  DAPDPS,    DAPDRO,    DAPDMU,    DPEDV,     DPEDG,    DPEDH ,        ORBIT
     *DPEDM     ,DPEDPS    ,DPECRO    ,DPEDMU    ,DANDV     ,DANDG     ,    ORBIT
     *  DANDH ,    DANDM ,    DANDPS,    DANDRO,    DANDMU,   DCXDV ,        ORBIT
     *  DCXDG ,    DCXDH ,    DCXDM ,    DCXDPS,    DCXDRO,   DCXDMU,        ORBIT
     *  DCYDV ,    DCYDG ,    DCYDH ,    DCYDM ,    DCYDPS,   DCYDRO,        ORBIT
     *  DCYDMU,    DASDV ,    DASDG ,    DASDH ,    DASDM ,   DASDPS,        ORBIT
     *  DASDRO,    DASDMU,    DENDV ,    DENDG ,    DENDH ,   DENDM ,        ORBIT
     *  DENDPS,    DENDRO,    DENDMU,    DMODV ,    DMODG ,   DMODH ,        ORBIT
     *  DMODM ,    DMODPS,    DMODRO,    DMODMU                              ORBIT
      DIMENSION ORBPRM(18),PPO(7,18)                                         ORBIT
      EQUIVALENCE (VI,ORBPRM),(DVIDV,PPO)                                    ORBIT
      COMMON/ORBIT/ YMXRF,SNXLMR,CSXLMR,SDOWN,SCROSS,TD,TC                   ORBIT
     *,SNPSR     ,CSPSR     ,SNGI      ,CSGI      ,SPSII     ,CPSII     ,    ORBIT
     *,STOT      ,CSI       ,SNI       ,SNGNU     ,CSANO     ,COSDMU    ,    ORBIT
     *,SINDMU    ,THT       ,WTFUEL                                          ORBIT
      COMMON/GLOBAL/                                                         GLOBAL
     *GR     ,ER     ,OMGZ   ,XLAMRF,YMURF ,LUM    ,TG     ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)        ,IFATAL,NARC   ,NBRAN ,NFARC ,ID(4) ,KTAB(20),   GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,            GLOBAL
     *INEOFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                               GLOBAL
C        I       INITIAL  STATE  SAVE                                        JUL21
      ENTRY QINTRI                                                           JUL21
      RAD = 57.2957795130823                                                 JUL21
      FTNM = 1./6076.10333                                                   JUL21
      IF(JTYP.NE.2) GO TO 5                                                  JUL21
      IPFLG1=0                                                               JUL21
      IPFLG2=0                                                               JUL21
      IPFLG3=0                                                               JUL21
    5 CONTINUE                                                               JUL21
      SQ(17, 1) = Y(1)                                                       JUL21
      SQ(17, 2) = Y(2)*RAD                                                   JUL21
      SQ(17, 3) = Y(4)-ER                                                    JUL21
      SQ(17, 5) = Y(5)*RAD                                                   JUL21
      SQ(17, 4) = Y(3)*RAD                                                   JUL21
      SQ(18, 1)=Y(6)*RAD                                                     JUL21
      NORBI = ORBI                                                           JUL21
      MORBI = 18*(NORBI - 1) + 7                                             JUL21
      SV(7) = GR*Y(MORBI)                                                    JUL21
      IF(NFARC .EQ. NARC) GO TO 7                                            JUL21
      INTB = 2                                                               JUL21
      NSB = NBRAN                                                            JUL21
      NSAB = NFARC - NSB                                                     JUL21
      GO TO 8                                                                JUL21
    7 INTB = 0                                                               JUL21
    8 CONTINUE                                                               JUL21
      IF(INTB.NE.2) RETURN                                                   JUL21
      SV(21)=1.                                                              JUL21
      CALL QLNVPI                                                            JUL21
      RETURN                                                                 JUL21
C        II       INTERMEDIATE  ARC DATA                                     JUL21
      ENTRY QLAEND                                                           JUL21
      IM1 = IARC                                                             AUG15
      IF(JTYP.LE.0) GO TO 30                                                 JUL21
C        II-A   PHASE 1  SIZING  DATA                                        JUL21
   10 CONTINUE                                                               JUL21
C  II-B TEST  FOR  KEY  ARCS                                                 JUL21
C                 BOOSTER  THRUST TERMINATION ARC                            JUL21
      IF   (IFIX(SQ(1,1)).EQ.IM1) GO TO 100                                  JUL21
C                 OPTIMAL  STAGE TIME                                        JUL21
   20 IF( IFIX(SQ(1 ,2)).EQ.IM1)  GO TO 110                                  JUL21
```

```
C               OPTIMAL  PITCHOVER TIME                                  JUL21
     30 IF  (IFIX(SQ(13,3)).EQ.IM1) GO TO  120                          JUL21
C              TEST  FOR  BRANCHING  AND  INJECTION                      JUL21
        IF(INTB.EQ.2.AND.NSB+NSAB.GE.IFIX(SQ(1,3)).AND.IARC.EQ.NSB+NSAB) JUL21
       * GO  TO 500                                                      JUL21
C              TEST  FOR   BRANCHING  AND  ENTRY TERMINUS                JUL21
        IF(INTB.EQ.2.AND.NSB+NSAB.LT.IFIX(SQ(1,3)).AND.IARC.EQ.NSB+NSAB) JUL21
       *GO  TO  600                                                      JUL21
        RETURN                                                           JUL21
C              BOOSTER   CHARACTERISTIC  VELOCITY                        JUL21
    100 VSTG   =   AP(17)                                                JUL21
        IF(JTYP.EQ.1) GO TO 20                                           JUL21
C         III-A  PHASE II   AT  STAGING  POINT                           JUL21
        QP(9) =AP(9)                                                     JUL21
        SW(10)=STOT/ER                                                   JUL21
        SW(20)=AP(1)                                                     JUL21
        SV(7)=W                                                          JUL21
        SV(8) =AP(3)                                                     JUL21
        SV(9) =AP(2)                                                     JUL21
        SV(10)=AP(4)                                                     JUL21
        SV(12)=AP(32)                                                    JUL21
        SV(17)=AP(5)                                                     JUL21
        SV(18)=AP(7)                                                     JUL21
        SV(19)=AP(6)                                                     JUL21
        SV(21)= 0.                                                       JUL21
        SV(22)= R                                                        JUL21
        SV(23)= AP(30)                                                   JUL21
        SV(24)= AP(29)                                                   JUL21
        SV(25)= 0.                                                       JUL21
        SV(26)= 90.- AP(92)                                              JUL21
        SQ(12,1) = AP(77)/FTNM/ER                                        JUL21
        SQ(36,1)= AP(55)                                                 JUL21
        SQ(36,2)= AP(56)                                                 JUL21
        SQ(37,1)= AP(1)                                                  JUL21
        GO TO 20                                                         JUL21
    110 SQ(18,2) = AP(8)                                                 JUL21
        GO TO 150                                                        JUL21
    120 SQ(18,3) = AP(8)                                                 JUL21
    150 RETURN                                                           JUL21
    300 CONTINUE                                                         JUL21
        RETURN                                                           JUL21
        ENTRY TRJNDQ                                                     JUL21
        IF(JTYP-1)150, 310,400                                           JUL21
    310 IOVEL= AP(17)                                                    JUL21
        DVO  = AP(17) -VSTG                                              JUL21
        WFO  = AP(9)                                                     JUL21
        GO TO 150                                                        JUL21
    400 CONTINUE                                                         JUL21
        IF(INTB.NE.2) GO TO 500                                          JUL21
        IF(IFIX(SQ(1,3))-NSB-NSAB)600,500,500                            JUL21
C            PHASE II  INJECTION PT.  DATA                               JUL21
    500 PZ(1) = AP(7)                                                    JUL21
        PZ(2) = AP(6)                                                    JUL21
        PZ(3) = AP(2)                                                    JUL21
        PZ(4) = Y(3)*RAD - AP(5)                                         JUL21
        PZ(5) = AP(20)                                                   JUL21
        SV(7) = SQ(3,5)                                                  JUL21
        SV(3) = AP(17)                                                   JUL21
        SV(4) = AP(9)                                                    JUL21
        SV(5) = SV(7) -SV(4)                                             JUL21
        SQ(3,1)=  AP(67)                                                 JUL21
        SQ(3,2)=  AP(68)                                                 JUL21
        SQ(3,3)=  AP(69)                                                 JUL21
        SQ(14,4)= AP(10)                                                 JUL21
        SQ(14,5)= AP(2)                                                  JUL21
        SQ(15,3)= AP(11)                                                 JUL21
        SQ(15,4)= AP(64)                                                 JUL21
        SQ(36,3)= AP(55)                                                 JUL21
        SQ(36,4)= AP(56)                                                 JUL21
        SQ(36,5)= AP(57)                                                 JUL21
C          ORBITER  BURN TIME                                            JUL21
```

```
      IORBI = SQ(1,3)                                                  JUL21
      SQ(37, 2) = AP(1) - SQ(37, 1)                                    JUL21
      RETURN                                                           JUL21
C        PHASE II  ENTRY  END POINT  (CRUISE RANGE)                    JUL21
  600 SW(15) = AP(20)                                                  JUL21
      RETURN                                                           JUL21
      END                                                              JUL21
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DVO		O	Orbiter ideal velocity (fps)	/SIZING/(	307)	QLTOSZ	O	DVO
ER	E_R	I	Earth radius. (FT)	/GLOBAL/(	2)	ENVPRQ	I	ER
						PDBCQL	I	ER
						QLTOSZ	I	ER
						STATEF	I	ER
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCQL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
IARC	I	I	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IDVEL		O	Total ideal velocity required to orbit (fps)	/SIZING/(	297)	QLTOSZ	O	IDVEL
IPFLG1		O	IPFLG1≠0 supresses print-out of velocity losses and inertial Euler angles.	/GLOBAL/(	69)	OUTPUT	I	IPFLG1
						PDBCQL	I	IPFLG1
						QLTOSZ	O	IPFLG1
						SALVE	I	IPFLG1
IPFLG2		O	IPFLG2≠0 supresses print-out of orbital parameters.	/GLOBAL/(	70)	QLTOSZ	O	IPFLG2
IPFLG3		O	IPFLG3≠0 supresses print-out of impact data.	/GLOBAL/(	71)	OUTPUT	I	IPFLG3
						QLTOSZ	O	IPFLG3
JTYP		I	Sizing. Flag.	/SIZING/(	313)	QLTOSZ	I	JTYP
						WRAPUP	I	JTYP
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND	I	NARC
						BNDRY	I	NARC
						CHECK	I	NARC
						ENDPT	I	NARC
						ENVPRQ	I	NARC
						FETCH	I	NARC
						INARC	I	NARC
						MAGIC	I	NARC
						QLTOSZ	I	NARC
						SALVE	I	NARC
						WRAPUP	I	NARC
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY	I	NBRAN
						BRANPT	I	NBRAN
						COSTAB	I	NBRAN
						ENVPRQ	I	NBRAN
						INTRPT	I	NBRAN
						MAGIC	I	NBRAN
						QLTOSZ	I	NBRAN
						SALVE	I	NBRAN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRQ	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						QLTOSZ	I	NFARC
						SALVE	I	NFARC
ORBI		I	Orbiter ignition arc	/SIZING/(	316)	QLTOSZ	I	ORBI
R	R	I	Radial distance from earth center to vehicle (FT)	/DYNA /(	7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRQ	I	R
						NLDRV	I	R
						PDBCQL	I	R
						QLTOSZ	I	R
						STATEF	M	R
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRQ	M	SQ
						QLTOSZ	M	SQ
STOT	S_T	I	Total range (FT)	/ORBIT /(	158)	OUTPUT	I	STOT
						PDBCQL	O	STOT
						QLTOSZ	I	STOT
SV		M	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRQ	M	SV
						QLTOSZ	M	SV
SW		O	A synthesis array (20) containing counters and sizing options	/SIZING/(	26)	QLTOSZ	O	SW
VSTG		M	Booster staging velocity (fps)	/SIZING/(	311)	QLTOSZ	M	VSTG
W	W	I	Weight (LBS)	/DYNA /(	91)	AL5	I	W
						ENVPRQ	I	W
						OUTPUT	I	W
						PDBCQL	I	W
						QLTOSZ	I	W
						STATEF	M	W
						TH3	I	W
WFO		O	Orbiter burnout weight (lb)	/SIZING/(	296)	QLTOSZ	O	WFO
Y		I	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y /(	1)	GROPE	O	Y
						INARC	M	Y
						MADAMS	M	Y
						QLTOSZ	I	Y
						RKUTTI	M	Y
						SALVE	M	Y
						WRAPUP	I	Y

SUBRØUTINE
RKUTT1

Purpose

RKUTT1 carries out the standard fourth order Runge-Kutta integration of the particular and homogeneous solutions. It is used over the first three intervals of each subarc as a starting procedure for MADAMS.*

*See Section 17.6 of Vol. I.

```
      SUBROUTINE RKUTT1(F)                                              RKUTT1
C                                                                       RKUTT1
C                   THIS ROUTINE CARRIES OUT THE STANDARD FOURTH ORDER  RKUTT1
C                   RUNGE-KUTTA INTEGRATION OF THE PARTICULAR AND HOMO- RKUTT1
C                   GENEOUS SOLUTIONS.  IT IS USED OVER THE FIRST THREE RKUTT1
C                   INTERVALS OF EACH SUBARC AS A STARTING PROCEDURE    RKUTT1
C                   FOR MADAMS.                                         RKUTT1
C                                                                       RKUTT1
      COMMON /S/ S                                                      RKUTT1
      COMMON /Z/ Z(50)                                                  Z
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      COMMON /ZI/ ZI(20, 4)                                             ZI
      COMMON /Y/ Y(820)                                                 Y
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QI(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /STUFF/ FK(820, 4)                                         RKUTT1
      DIMENSION S(820),CC(4), FS(1), F(1)                               RKUTT1
      EQUIVALENCE (FS, FK)                                              RKUTT1
C                                                                       RKUTT1
      T = H/2.                                                          RKUTT1
      DATA CC/ 1., 2., 2., 1./                                          RKUTT1
      DO 1 I = 1, NN                                                    RKUTT1
    1 FS(I) = F(I)                                                      RKUTT1
      X = X + T                                                         RKUTT1
      CALL INTERP                                                       RKUTT1
      DO 5 J = 2, 4                                                     RKUTT1
      IF(J .NE. 4) GO TO 3                                              RKUTT1  3
      X = X + T                                                         RKUTT1
      T = H                                                             RKUTT1
      IF(KONVER) GO TO 3                                                RKUTT1  3
      NEWNOM = .TRUE.                                                   RKUTT1
      L = KPT + 1                                                       RKUTT1
      DO 2 I = 1, N                                                     RKUTT1
    2 Z(I) = ZI(I, L)                                                   RKUTT1
    3 L = J - 1                                                         RKUTT1
      DO 4 I = 1, NN                                                    RKUTT1
    4 S(I) = Y(I) + T*FK(I, L)                                          RKUTT1
    5 CALL LINDRV(S, FK(1, J))                                          RKUTT1
      DO 7 J = 1, 4                                                     RKUTT1
      T = CC(J)/6.*H                                                    RKUTT1
      DO 7 I = 1, NN                                                    RKUTT1
    7 Y(I) = Y(I) + T*FK(I, J)                                          RKUTT1
      RETURN                                                            RKUTT1
      END                                                               RKUTT1
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
FK		I	An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-2 thru -5 in Vol.I of this document.	/STUFF	/(1)	MADAMS	M	F1
						RKUTT1	I	FK
						RKUTT1	O	FS
FS		O	An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-2 thru -5 in Vol.I of this document.	/STUFF	/(1)	MADAMS	M	F1
						RKUTT1	I	FK
						RKUTT1	O	FS
H	h	I	Integration step size in quasitime.	/D	/(2)	AL4	I	H
						INARC	M	H
						MADAMS	I	H
						RKUTT1	I	H
						RKUTT2	I	H
						SALVE	M	H
						WRAPUP	M	DT
KONVER		I	Logical flag that indicates to the QL module that the QL iteration is converged.	/CNTRL	/(28)	ALGCON	I	KONVER
						APPLY	I	KONVER
						ARCIN	I	KONVER
						COHOMO	O	KONVER
						GROPE	O	KONVER
						NLDRV	I	KONVER
						OUTPUT	I	KONVER
						RKUTT1	I	KONVER
KPT		I	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL	/(8)	BCOND	O	KPT
						BNDRY	O	KPT
						FORCES	I	KPT
						MAGIC	O	KPT
						RKUTT1	I	KPT
						SALVE	M	KPT
						WRAPUP	M	KPT
N		I	Total number of QL state and costate variables. N = 18.	/PC	/(2)	BNDRY	I	N
						CHECK	I	N
						INARC	I	N
						LINDRV	I	N
						NLDRV	I	N
						NOMNAL	I	N
						RKUTT1	I	N
						SALVE	I	N
						WRAPUP	I	N
NEWNOM		O	A logical flag that indicates to the Runge-Kutta integration whether or not the system Jacobian needs to be reevaluated.	/CNTRL	/(15)	INTERP	O	NEWNOM
						LINDRV	M	NEWNOM
						RKUTT1	O	NEWNOM
						SALVE	O	NEWNOM
						WRAPUP	O	NEWNOM
NN		I	The number of quantities currently being numerically integrated.	/CNTRL	/(52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NOMNAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
S		M	An 820 word array used to store the particular and homogeneous solutions.	/S	/(1)	NOMNAL	M	S
						RKUTT1	M	S
X	x	M	The quasitime variable.	/D	/(1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
Y		M	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y	/(1)	GROPE	O	Y
						INARC	M	Y
						MADAMS	M	Y
						QLTOSZ	I	Y
						RKUTT1	M	Y
						SALVE	M	Y
						WRAPUP	I	Y
Z	Z	O	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRQ	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NORMAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZI		I	A 20x4 array containing the first four values of Z in the present subarc.	/ZI	/(1)	INTERP	I	ZI
						RKUTT1	I	ZI
						SALVE	M	ZI

SUBRØUTINE RKUTT2

Purpose

RKUTT2 carries out the standard fourth order Runge-Kutta integration of the converged state/costate solution and the velocity losses.*

---

*See Section 17.6 in Vol. I.

```
      SUBROUTINE RKUTT2                                                 RKUTT2
C                                                                       RKUTT2
C                THIS ROUTINE CARRIES OUT THE STANDARD FOURTH-ORDER     RKUTT2
C                RUNGE-KUTTA INTEGRATION OF THE CONVERGED STATE/CO-     RKUTT2
C                STATE SOLUTION AND THE VELOCITY LOSSES                 RKUTT2
C                                                                       RKUTT2
      COMMON /Z/ Z(50)                                                  Z
      COMMON /ZD/ ZD(50)                                                ZD
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER   ,ITAPA ,ITAPB ,JMIN   ,JMAX  ,LINES ,KPT   ,MOM   ,  CNTRL
     *KARD  ,INDX(4)       ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,  CNTRL
     *KPAGE ,NNP    ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,  CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      DIMENSION FK(50, 4), S(50), CC(4), FS(1)                          RKUTT2
      EQUIVALENCE (FS, FK)                                              RKUTT2
C                                                                       RKUTT2
      DATA CC/ 1., 2., 2., 1./                                          RKUTT2
      T = H/2.                                                          RKUTT2
      DO 1 I = 1, NN                                                    RKUTT2
    1 FS(I) = ZD(I)                                                     RKUTT2
      X = X + T                                                         RKUTT2
      DO 5 J = 2, 4                                                     RKUTT2
      IF(J .NE. 4) GO TO 3                                              RKUTT2
      X = X + T                                                         RKUTT2
      T = H                                                             RKUTT2
    3 L = J - 1                                                         RKUTT2
      DO 4 I = 1, NN                                                    RKUTT2
    4 S(I) = Z(I) + T*FK(I, L)                                          RKUTT2
    5 CALL NLDRV(S, FK(1, J))                                           RKUTT2
      DO 7 J = 1, 4                                                     RKUTT2
      T = CC(J)/6.*H                                                    RKUTT2
      DO 7 I = 1, NN                                                    RKUTT2
    7 Z(I) = Z(I) + T*FK(I, J)                                          RKUTT2
      RETURN                                                            RKUTT2
      END                                                               RKUTT2
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
H	h	I	Integration step size in quasitime.	/D	/(2)	AL4	I	H
						INARC	M	H
						MADAMS	I	H
						RKUTT1	I	H
						RKUTT2	I	H
						SALVE	M	H
						WRAPUP	M	DT
NN		I	The number of quantities currently being numerically integrated.	/CNTRL	/(52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NOMNAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
X	x	M	The quasitime variable.	/D	/(1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
Z	Z	M	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRO	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZD		I	A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.I of this document.	/ZD	/(1)	ENVPRO	I	ZD
						LINDRV	I	ZD
						OUTPUT	I	ZD
						RKUTT2	I	ZD
						WRAPUP	I	ZD

SUBRØUTINE
SALVE

Purpose

SALVE controls the subarc by subarc integration of the particular and homogeneous solutions. In addition it sets up all initial conditions on the state and costate.*

*See Section 17 of Vol. I.

SALVE

```
      PROGRAM SALVE                                                     SALVE
C                                                                       SALVE
C                    THIS ROUTINE CONTROLS THE INTEGRATION OF THE PARTI-  SALVE
C                    CULAR AND HOMOGENEOUS SOLUTIONS OVER ALL THE SUBARCS SALVE
C                                                                       SALVE
      COMMON /S/ S(820)                                                 SALVE
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,    CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,    CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,    CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON /JACOB/ JAKE(400)                                          JACOB
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF     ,EJ       ,XISP     ,TMULT    ,DTNC     ,DTPI     ,     ARCDAT
     *IATM     ,IMODE    ,JAER     ,JPRO     ,QMAX     ,GMAX     ,     ARCDAT
     *XLMAX    ,HDMAX    ,GMDOT    ,ALFMAX   ,PHMAX    ,MAEA     ,     ARCDAT
     *MAEB     ,MAEC     ,MAED     ,MAEE     ,MAEF     ,MAEG     ,     ARCDAT
     *MT       ,MISP     ,MXCG     ,MZCG     ,MWDA     ,MWDB     ,     ARCDAT
     *MDB      ,XCGR     ,ZCGR     ,XE       ,ZE       ,XT       ,     ARCDAT
     *DREF     ,MCND     ,RHOB     ,QMULT    ,REMAX    ,FRATE          ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,    GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20),  GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,        GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), DT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /Z/ Z(50)                                                  Z
      COMMON /ZI/ ZI(20, 4)                                             ZI
      COMMON /Y/ Y(820)                                                 Y
      COMMON /F/ F(820, 4)                                              F
      COMMON /PC/                                                       PC
     *PC1   ,N     ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX      PC
      DIMENSION BNOM(18), CNOM(18)                                      SALVE
      L = 0                                                             SALVE
      DO 14 IARC = 1, NARC                                              SALVE
      DO 90 I = 1, 400                                                  SALVE
   90 JAKE(I) = 0                                                       SALVE
C                                                                       SALVE
C     READ IN THE DATA FOR THIS SUBARC.                                 SALVE
C                                                                       SALVE
      CALL READMS(9, ARCDA, 42, IARC)                                   SALVE
C                                                                       SALVE
C     STORE THE NUMBER OF POINTS FOR THIS SUBARC AND COMPUTE THE STEP-  SALVE
C     SIZE.                                                             SALVE
C                                                                       SALVE
      NPTS = NPOINT(IARC)                                               SALVE
      H = 1. 0/FLOAT(NPTS - 1)                                          SALVE
C                                                                       SALVE
C     COMPUTE NUMBER OF HOMOGENEOUS SOLUTIONS FOR THIS SUBARC.          SALVE
C                                                                       SALVE
      MOM =        NOC(IARC)                                            SALVE
      NN = N*(MOM + 1)                                                  SALVE
C                                                                       SALVE
C     STORE THE FIRST FOUR POINTS OF THIS SUBARC.                       SALVE
C                                                                       SALVE
      X = IARC - 1                                                      SALVE
      NOPRNT = .FALSE.                                                  SALVE
      DO 9 KPT = 1, 4                                                   SALVE
      INDX(KPT) = KPT                                                   SALVE
      XI(KPT) = X                                                       SALVE
      CALL NOMNAL                                                       SALVE
      DO 8 I = 1, N                                                     SALVE
```

```
      8 ZI(I, KPT) = Z(I)                                                SALVE
      9 X = X + H                                                        SALVE
C                                                                        SALVE
C     RESTORE THE FIRST POINT OF THE NOMINAL.                            SALVE
C                                                                        SALVE
      DO 10 I = 1, N                                                     SALVE
   10 Z(I) =ZI(I, 1)                                                     SALVE
      X = IARC - 1                                                       SALVE
      NEWNOM = .TRUE.                                                    SALVE
C                       SET UP THE INITIAL CONDITIONS OF THE PARTICULAR  SALVE
C                       AND HOMOGENEOUS SOLUTIONS FOR THE START OF THIS  SALVE
C                       ARC                                              SALVE
      DO 210 I = 1, 9                                                    SALVE
      IF(IIC(I, IARC) - 1) 201, 202, 204                                 SALVE
C                       CONTINUOUS STATE.  IF THIS IS THE START OF THE 2ND SALVE
C                       BRANCH, SET PARTICULAR SOL. TO VALUE AT END OF STEM. SALVE
C                       OTHERWISE, LET PART. SOL. GO ACROSS CONTINUOUSLY. SALVE
  201 IF(IARC .EQ. NFARC + 1) Y(I) = BNOM(I)                             SALVE
      GO TO 210                                                          SALVE
  202 DO 203 J = I, NN, 18                                               SALVE
C                       KNOWN STATE. ZERO THIS STATE*S ROW IN ALL HOMOGEN- SALVE
C                       EOUS SOLUTIONS UP TO THIS PT. AND SET THE PART. SOL. SALVE
C                       TO THE DESIRED VALUE.                            SALVE
  203 Y(J) = 0                                                           SALVE
      Y(I) = VALIC(I, IARC)                                              SALVE
      GO TO 210                                                          SALVE
  204 IF(IIC(I, IARC) - 5) 205, 206, 208                                 SALVE
C                       UNKNOWN STATE.  SET THE PART. SOL. FOR THIS STATE SALVE
C                       EQUAL TO THE TOTAL SOL. FROM LAST ITER. AND INTRO- SALVE
C                       DUCE A NEW HOMO. SOL.                            SALVE
  205 Y(I) = Z(I)                                                        SALVE
      L = L + 18                                                         SALVE
      IL = L + I                                                         SALVE
      Y(IL) = 1                                                          SALVE
      GO TO 210                                                          SALVE
C                       KNOWN DROP WEIGHT.  IS THIS THE START OF 2ND BRANCH SALVE
  206 IF(IARC .EQ. NFARC + 1) GO TO 207                                  SALVE
C                       NO.  SUBTRACT OFF DROP WEIGHT FROM PART. SOL.    SALVE
      Y(I) = Y(I) - VALIC(I, IARC)                                       SALVE
      GO TO 210                                                          SALVE
C                       START OF 2ND BRANCH.  SUBTRACT DROP WT. FROM PART. SALVE
C                       SOL. AT END OF STEM.                             SALVE
  207 Y(I) = BNOM(I) - VALIC(I, IARC)                                    SALVE
      GO TO 210                                                          SALVE
C                       SIZING DROP WT.  IS THIS THE START OF 2ND BRANCH SALVE
  208 IF(IARC .EQ. NFARC + 1) GO TO 209                                  SALVE
C                       NO.  SUBT. DROP WT. FROM PART. SOL.              SALVE
      WPRO = GR*(ZSAVE(7) - M)                                           SALVE
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                   SALVE
      Y(I) = Y(I) - WDRP/GR                                              SALVE
      GO TO 210                                                          SALVE
  209 IF(IIC(I, IARC) .LT. 7) GO TO 2091                                 SALVE
C                       WT. DISTRIBUTION BETWEEN STEM AND BRANCHES. SUBT. SALVE
C                       INITIAL WT. AT START OF 1ST BRANCH FROM WT. AT END SALVE
C                       OF STEM.                                         SALVE
      Y(7) = BNOM(7) - CNOM(7)                                           SALVE
      GO TO 210                                                          SALVE
C                       SUBT. SIZING DROP WT. FROM THE WT. AT END OF STEM. SALVE
 2091 WPRO = GR*(ZSAVE(7) - BNOM(7))                                     SALVE
      CALL WTDRP(WPRO, WDRP, DWDRP, 3)                                   SALVE
      Y(I) = BNOM(7) - WDRP/GR                                           SALVE
  210 CONTINUE                                                           SALVE
      DO 214 I = 1, 9                                                    SALVE
      IF(IICT(I, IARC) -1) 214, 211, 213                                 SALVE
C                       KNOWN COSTATE.  ZERO OUT THE APPROPRIATE ROW OF  SALVE
C                       PART.AND HOMO. SOLS.                             SALVE
  211 DO 212 J = I, NN, 18                                               SALVE
  212 Y(J + 9) = 0                                                       SALVE
      GO TO 214                                                          SALVE
C                       IS THIS THE START OF THE 2ND   BRANCH            SALVE
```

```
  213 IF(IARC .NE. NFARC + 1) GO TO 2131                             SALVE
C                   YES.  IS THIS COSTATE UNKNOWN.                   SALVE
      IF(IICT(I, IARC) .EQ. 2) GO TO 2131                            SALVE
C                   COSTATE DISTRIBUTION FROM STEM TO BRANCHES.SET PART. SALVE
C                   SOL. FOR THIS COSTATE EQUAL TO DIFF.BETWEEN END OF   SALVE
C                   STEM AND START OF 1ST ARC                        SALVE
      Y(I + 9) = BNOM(I + 9) - CNOM(I + 9)                           SALVE
      GO TO 214                                                      SALVE
 2131 Y(I + 9) = Z(I + 9)                                            SALVE
C                   UNKNOWN COSTATE. SET PART. SOL. TO VALUE OF TOTAL SALVE
C                   SOL. FROM LAST ITER. AND INTRODUCE A NEW HOMO SOL. SALVE
      L = L + 18                                                     SALVE
      IL = L + I                                                     SALVE
      Y(IL + 9) = 1                                                  SALVE
  214 CONTINUE                                                       SALVE
      IF(NFARC .EQ. NARC  .OR. IARC .EQ. NBRAN + 1) GO TO 216        SALVE
      DO 215 I = 1, 18                                               SALVE
  215 CNOM(I) = Y(I)                                                 SALVE
  216 CONTINUE                                                       SALVE
C                                                                    SALVE
C     START THE INTEGRATION BY RUNGE-KUTTA.                          SALVE
C                                                                    SALVE
      DO 12 KPT = 1, 4                                               SALVE
      IF(KPT .EQ. NPTS) NEWNOM = .TRUE.                              SALVE
      CALL LINDRV(Y, F(1, KPT))                                      SALVE
      WRITE(ITAPB) (Y(IJ), IJ = 1, NN)                               SALVE
      IF(KPT .NE. 1) GO TO 121                                       SALVE
      CALL OUTPUT                                                    SALVE
      CALL WRITMS(41, Y, NN, 2*IARC - 1)                             SALVE
  121 IF(KPT .LT. 4) CALL RKUTT1(F(1, KPT))                          SALVE
C                   FINISH INTEGRATION OVER THIS ARC BY ADAMS-MOULTON. SALVE
   12 CONTINUE                                                       SALVE
      IF(KPT .EQ. NPTS)GO TO 131                                     SALVE
   13 KPT = KPT + 1                                                  SALVE
      NEWNOM = .TRUE.                                                SALVE
      CALL NOMNAL                                                    SALVE
      CALL MADAMS                                                    SALVE
      WRITE(ITAPB) (Y(IJ), IJ = 1, NN)                               SALVE
      IF(KPT .LT. NPTS) GO TO 13                                     SALVE
  131 CALL OUTPUT                                                    SALVE
      CALL WRITMS(41, Y, NN, 2*IARC)                                 SALVE
      IF(IARC .NE. NBRAN) GO TO 14                                   SALVE
C                   THIS IS THE END OF THE STEM OF A BRANCH PROBLEM. SALVE
C                   STORE THE PARTICULAR SOLUTION.                   SALVE
      DO 132 I = 1, 18                                               SALVE
  132 BNOM(I) = Y(I)                                                 SALVE
   14 CONTINUE                                                       SALVE
      IF(IPFLG1 .EQ. 0) CALL PRORPA                                  JUL19B
      RETURN                                                         SALVE
      END                                                            SALVE
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ARCDA	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
						BNDRY	I	ARCDA
						CHECK	I	ARCDA
						FETCH	I	ARCDA
						SALVE	I	ARCDA
						STATEF	I	SREF
						UT	I	SREF
						WRAPUP	I	ARCDA
F		I	An 820x4 array used to store the vectors k_1, k_2, k_3, and k_4 defined by Equations 17.6-7 thru -10 of Vol.I of this document.	/F /(	1)	MADAMS	I	F
						SALVE	I	F
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCOL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
H	h	M	Integration step size in quasitime.	/D /(	2)	AL4	I	H
						INARC	M	H
						MADAMS	I	H
						RKUTT1	I	H
						RKUTT2	I	H
						SALVE	M	H
						WRAPUP	M	DT
IARC	I	M	Subarc number.	/CNTRL /(	24)	ARCIN	I	IARC
						BCOND	M	IARC
						BNDRY	M	IARC
						BRANPT	I	IARC
						CHECK	M	IARC
						COSTAB	I	IARC
						COSTAI	I	IARC
						ENDPT	I	IARC
						FORCES	I	IARC
						INARC	M	IARC
						INTRPT	I	IARC
						MAGIC	M	IARC
						MARCH	I	IARC
						QLTOSZ	I	IARC
						SALVE	M	IARC
						WRAPUP	M	IARC
IIC		I	A 10x20 array containing the initial condition codes for the QL state vector. The columns correspond to subarc starting points, the rows, to QL state variables.	/BLOCK /(	1)	BCOND	M	IIC
						BRANPT	I	IIC
						CHECK	I	IIC
						COSTAB	I	IIC
						COSTAI	I	IIC
						COSTAO	I	IIC
						INTRPT	I	IIC
						SALVE	I	IIC
IICT		I	A 10x20 array containing the target condition codes for the state vector. The columns correspond to the subarc end points. The nonzero entries in a column are the state target condition codes that apply at the end of the corresponding subarc.	/BLOCK /(	201)	CHECK	I	IICT
						COSTAB	M	IICT
						COSTAI	M	IICT
						COSTAO	O	IICT
						MAGIC	O	IICT
						SALVE	I	IICT
INDX		O	An array of four words that indicate to Adams-Moulton integration in what order the derivatives of the particular and homogeneous solutions are stored.	/CNTRL /(	11)	BCOND	M	INDX
						MADAMS	M	INDX
						SALVE	O	INDX
IPFLG1		I	IPFLG1≠0 supresses print-out of velocity losses and inertial Euler angles.	/GLOBAL/(	69)	OUTPUT	I	IPFLG1
						PDBCOL	I	IPFLG1
						QLTOSZ	O	IPFLG1
						SALVE	I	IPFLG1

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
ITAPB		I	Number of the logical unit onto which the quasitime histories of the particular and homogeneous solutions from the current OL iteration are written.	/CNTRL /(	4)	GROPE SALVE	O I	ITAPB ITAPB
JAKE		O	An 18x18 array defined by Equation 17.5-5 in Vol.I of this document. The entry in the i-th row and j-th column is the total partial derivative of the quasitime derivative of the Y_i component of Y with respect to the Y_j component of Y, i.e., $\partial Y_i'/\partial Y_j$, where $Y^T = (y^T, \lambda^T)$	/JACOB /(	1)	LINDRV NLDRV NLDRV SALVE	I M M O	JAKE JACOB VDV JAKE
KPT		M	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND BNDRY FORCES MAGIC RKUTTI SALVE WRAPUP	O O I O I M M	KPT KPT KPT KPT KPT KPT KPT
M	m	I	Mass (G'S)	/D /(	97)	AL4 AL7 AL8 AL9 APPLY BRANPT COSTAB COSTAI INTRPT NLDRV OUTPUT SALVE STATEF WRAPUP	I I I I I I I I I I I I I I	M M M M M M M M M M M M M M
MOM		M	The number of homogeneous solutions currently being integrated.	/CNTRL /(	9)	GROPE INARC LINDRV NORMAL SALVE WRAPUP	O M I I M M	MOM MOM MOM MOM MOM MOM
N		I	Total number of OL state and costate variables. N = 18.	/PC /(	2)	BNDRY CHECK INARC LINDRV NLDRV NORMAL RKUTTI SALVE WRAPUP	I I I I I I I I I	N N N N N N N N N
NARC	N_3	I	Number of subarcs in the problem.	/GLOBAL/(	18)	BCOND BNDRY CHECK ENDPT ENVPRO FETCH INARC MAGIC OLTOSZ SALVE WRAPUP	I I I I I I I I I I I	NARC NARC NARC NARC NARC NARC NARC NARC NARC NARC NARC
NBRAN	N_1	I	Number of the last subarc on the stem of a branch problem. If the problem is not a branch problem, then NBRAN = 0.	/GLOBAL/(	19)	BNDRY BRANPT COSTAB ENVPRO INTRPT MAGIC OLTOSZ SALVE	I I I I I I I I	NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN NBRAN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
NEWNOM		O	A logical flag that indicates to the Runge-Kotta integration whether or not the system Jacobian needs to be reevaluated.	/CNTRL /(	15)	INTERP	O	NEWNOM
						LINDRV	M	NEWNOM
						RKUTT1	O	NEWNOM
						SALVE	O	NEWNOM
						WRAPUP	O	NEWNOM
NFARC	N_2	I	Number of the last subarc on the first branch. If the problem is not a branch problem, then NFARC = NARC.	/GLOBAL/(	20)	BCOND	I	NFARC
						BNDRY	I	NFARC
						BRANPT	I	NFARC
						COSTAB	I	NFARC
						ENVPRO	I	NFARC
						INTRPT	I	NFARC
						MAGIC	I	NFARC
						QLTOSZ	I	NFARC
						SALVE	I	NFARC
NN		M	The number of quantities currently being numerically integrated.	/CNTRL /(	52)	BNDRY	M	NN
						INARC	M	NN
						MADAMS	I	NN
						MAGIC	M	NN
						NOMNAL	I	NN
						RKUTT1	I	NN
						RKUTT2	I	NN
						SALVE	M	NN
						WRAPUP	M	NN
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.	/BLOCK /(	842)	BNDRY	I	NOC
						BRANPT	I	NOC
						COSTAB	O	NOC
						COSTAI	O	NOC
						COSTAG	O	NOC
						INARC	I	NOC
						INTRPT	I	NOC
						SALVE	I	NOC
						WRAPUP	I	NOC
NOPRNT		O	Not used.	/CNTRL /(	29)	OUTPUT	M	NOPRNT
						SALVE	O	NOPRNT
						TRAJIN	O	IFRNT
NPOINT		I	A twenty word array containing the number of points in each subarc.	/D /(	191)	INARC	O	NPOINT
						SALVE	I	NPOINT
NPTS		M	The total number of points in the subarc.	/CNTRL /(	19)	BCOND	O	NPTS
						BNDRY	O	NPTS
						FORCES	I	NPTS
						INARC	M	NPTS
						MAGIC	O	NPTS
						SALVE	M	NPTS
						WRAPUP	O	NPTS
VALIC		I	A 10x20 array containing the desired values of all the fixed (known) QL state variables. The columns correspond to the subarc starting points, the rows, to QL state variables.	/BLOCK /(	862)	BCOND	M	VALIC
						SALVE	I	VALIC
X	x	M	The quasitime variable.	/D /(	1)	AL4	I	X
						BNDRY	O	X
						ERROR	I	X
						FETCH	O	X
						FORCES	I	X
						INARC	M	X
						INTERP	I	X
						MADAMS	M	X
						RKUTT1	M	X
						RKUTT2	M	X
						SALVE	M	X
						STATEF	I	X
						WRAPUP	M	TT
XI		O	A four word array containing the first four values of quasitime in the subarc.	/D /(	3)	INTERP	I	XI
						SALVE	O	XI

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
Y		M	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.	/Y	/(1)	GROPE	O	Y
						INARC	M	Y
						MADAMS	M	Y
						QLTOS2	I	Y
						RKUTT1	M	Y
						SALVE	M	Y
						WRAPUP	I	Y
Z	Z	M	A 20 word array used to store the total linear solution from the preceding SL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRO	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINORJ	I	Z
						NORMAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZI		M	A 20x4 array containing the first four values of Z in the present subarc.	/ZI	/(1)	INTERP	I	ZI
						RKUTT1	I	ZI
						SALVE	M	ZI
ZSAVE		I	A twenty word array containing the values from the initial arc of the state and costate at the initial point of the trajectory.	/D	/(151)	BCOND	O	ZSAVE
						BRANPT	I	ZSAVE
						COSTAB	I	ZSAVE
						COSTAI	I	ZSAVE
						INTRPT	I	ZSAVE
						PDBCOL	I	ZSAVE
						SALVE	I	ZSAVE

SUBRØUTINE
SPLINE

Purpose

SPLINE interpolates the univariant tabular functions. In addition, it computes the first and second partials of these functions by evaluating the derivatives of the cubic spline interpolating function.*

*See Section 17.7 of Vol. I.

SPLINE

```
      SUBROUTINE SPLINE(IT, T, F, DFDX, D2FDX2)                         SPLINE
C                                                                       SPLINE
C                   THIS ROUTINE PERFORMS THE NATURAL CUBIC SPLINE      SPLINE
C                   INTERPOLATION AT THE PT  T OF THE FUNCTION WHICH    SPLINE
C                   IS TABULATED IN TABLE NO. IT.                       SPLINE
C                                                                       SPLINE
      EXTERNAL NUCASE                                                   SPLINE
      COMMON /TABLE/ TABLE(2100) /GLOBAL/ G(66)                         SPLINE
      DIMENSION X(1), Y(1), Z(1), LOCI(1), LOCL(1), LOCF(1)             SPLINE
      EQUIVALENCE (X, TABLE), (Y, TABLE(701)), (Z, TABLE(1401)),        SPLINE
     *(LOCI, X), (LOCL, Y), (LOCF, Z), (NT, G(66))                      SPLINE
      DATA SIXTH/17155252525252525253B/                                 SPLINE
      DATA SPLINT /6HSPLINE/                                            SPLINE
   10 FORMAT(1H0, 22HTHE TABLE NO. IN LOC. O6 17H IS OUT OF RANGE.)     SPLINE
   20 FORMAT(1H0, 22HTHE TABLE NO. IN LOC. O6 20H HAS NOT BEEN INPUT.)  SPLINE
C                   MAKE SURE TABLE NO. WITHIN RANGE.                   SPLINE
      IF(IT .LE. NT) GO TO 100                                          SPLINE
      ITLOC = XLOCF(IT)                                                 SPLINE
      WRITE(6, 10) ITLOC                                                SPLINE
      CALL ERROR(SPLINT, -1, 1)                                         SPLINE
      STOP                                                              SPLINE
  100 IF(IT .LE. 0) GO TO 110                                           SPLINE
C                   NON-ZERO TABLE NO. MAKE SURE IT WAS INPUT           SPLINE
      II = LOCI(IT)                                                     SPLINE
      IF(II .GT. 0) GO TO 120                                           SPLINE
      ITLOC = XLOCF(IT)                                                 SPLINE
      WRITE(6, 20) ITLOC                                                SPLINE
      CALL ERROR(SPLINT, -2, 1)                                         SPLINE
      STOP                                                              SPLINE
C                   ZERO TABLE NO.  SEND BACK CONSTANT ZERO FNC.        SPLINE
  110 F = 0.                                                            SPLINE
      DFDX = 0.                                                         SPLINE
      D2FDX2 = 0.                                                       SPLINE
      RETURN                                                            SPLINE
  120 IF = LOCF(IT)                                                     SPLINE
C                   WHAT KIND OF FUNCTION IS F.                         SPLINE
      IF(IF - II - 1) 130, 140, 150                                     SPLINE
C                   CONSTANT FNC.                                       SPLINE
  130 F = Y(II)                                                         SPLINE
      DFDX = 0.                                                         SPLINE
      D2FDX2 = 0.                                                       SPLINE
      RETURN                                                            SPLINE
C                   LINEAR FNC.                                         SPLINE
  140 Y1 = Y(II)                                                        SPLINE
      X1 = X(II)                                                        SPLINE
      DFDX=(Y(IF) - Y1)/(X(IF) - X1)                                    SPLINE
      F = Y1 + (T - X1)*DFDX                                            SPLINE
      D2FDX2 = 0.                                                       SPLINE
      RETURN                                                            SPLINE
C                   NONLINEAR FNC.  IS T.GT. FIRST TABLE ENTRY.         SPLINE
  150 IF(T .GT. X(II)) GO TO 160                                        SPLINE
C                   NO.  LINEARLY EXTRAP. FOR F(T)                      SPLINE
      Y1 = Y(II)                                                        SPLINE
      X1 = X(II)                                                        SPLINE
      IIP1 =  II + 1                                                    SPLINE
      DEL = X(IIP1) - X1                                                SPLINE
      DFDX=(Y(IIP1) - Y1)/DEL - SIXTH*DEL*Z(IIP1)                       SPLINE
      F = Y1 + (T - X1)*DFDX                                            SPLINE
      D2FDX2 = 0.                                                       SPLINE
      RETURN                                                            SPLINE
C                   IS T .LT. LAST ENTRY IN TABLE.                      SPLINE
  160 IF(T .LT. X(IF)) GO TO 170                                        SPLINE
C                   NO.  LINEARLY EXTRAP. FOR F(T)                      SPLINE
      IFM1 = IF - 1                                                     SPLINE
      Y1 = Y(IF)                                                        SPLINE
      X1 = X(IF)                                                        SPLINE
      DEL = X1 - X(IFM1)                                                SPLINE
      DFDX=(Y1 - Y(IFM1))/DEL+ SIXTH*Z(IFM1)*DEL                        SPLINE
      F = Y1 + (T - X1)*DFDX                                            SPLINE
      D2FDX2 = 0.                                                       SPLINE
      RETURN                                                            SPLINE
C                   FIND TABLE ENTRIES THAT BRACKET T.                  SPLINE
```

```
  170 IL = LOCL(IT)                                                   SPLINE
      IF(T - X(IL)) 180, 220, 190                                     SPLINE
  180 IL = IL - 1                                                     SPLINE
      IF(T - X(IL)) 180, 210, 210                                     SPLINE
  190 IS = IL + 1                                                     SPLINE
      DO 200 I = IS, IF                                               SPLINE
      IF(T - X(I  ) .LT. 0.) GO TO 210                                SPLINE
  200 IL = I                                                          SPLINE
  210 LOCL(IT) = IL                                                   SPLINE
  220 ILP1 = IL + 1                                                   SPLINE
C                    INTERPOLATE FOR F(T)                             SPLINE
      DX1 = T - X(IL)                                                 SPLINE
      DX2 = X(ILP1) - T                                               SPLINE
      Y1 = Y(IL)                                                      SPLINE
      Y2 = Y(ILP1)                                                    SPLINE
      Z1 = Z(IL)                                                      SPLINE
      Z2 = Z(ILP1)                                                    SPLINE
      DEL = DX1 + DX2                                                 SPLINE
      TEMP1= Z2*DX1 + Z1*DX2                                          SPLINE
      TEMP2 = SIXTH*DEL                                               SPLINE
      F = ((Y2*DX1 + Y1*DX2) + SIXTH*(Z2*DX1**3 + Z1*DX2**3))/DEL     SPLINE
     * - TEMP2*TEMP1                                                  SPLINE
      DFDX = ((Y2 - Y1) + .5*(Z2*DX1**2 - Z1*DX2**2))/DEL -TEMP2*(Z2-Z1) SPLINE
      D2FDX2 = TEMP1/DEL                                              SPLINE
      RETURN                                                          SPLINE
      END                                                             SPLINE
```

180 190 220
180 210
210

1343

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LOCF		I	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the last value of the independent variable of the corresponding table.	/TABLE /(	1401)	SPLINE SPLINE	I I	LOCF Z
LOCI		I	A 2100 word array used for storing up to 50 spline fitted univariant tables.	/TABLE /(	1)	SPLINE SPLINE	I I	LOCI X
LOCL		M	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that indicates the last interval in which interpolation of the corresponding table occured.	/TABLE /(	701)	SPLINE SPLINE	M I	LOCL Y
NT		I	Largest univariant table number in this case.	/GLOBAL/(	66)	SPLINE	I	NT
X		I	A 2100 word array used for storing up to 50 spline fitted univariant tables.	/TABLE /(	1)	SPLINE SPLINE	I I	LOCI X
Y		I	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that indicates the last interval in which interpolation of the corresponding table occured.	/TABLE /(	701)	SPLINE SPLINE	M I	LOCL Y
Z		I	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the last value of the independent variable of the corresponding table.	/TABLE /(	1401)	SPLINE SPLINE	I I	LOCF Z

SUBRØUTINE
STATEF

```
      SUBROUTINE STATEF                                                 STATEF
C                                                                       STATEF
C                   THIS ROUTINE COMPUTES ALL DYNAMIC QUANTITIES WHICH  STATEF
C                   BEAR NO EXPLICIT DEPENDENCE ON THE CONTROL          STATEF
C                                                                       STATEF
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,        GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,DMAX      ,GMAX      , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG~     ,MWDA      ,MWDB      , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACA ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION DYN(1)                                                  STATEF
      EQUIVALENCE (DYN, XX)                                             STATEF
      DATA XSTATF/6HSTATEF/                                             STATEF
      DATA DEG/57.2957795130823/                                        STATEF
      SINGAM = SIN(GAM)                                                 STATEF
      COSGAM = COS(GAM)                                                 JUL19B
      COS2GM = (COSGAM + SINGAM)*(COSGAM - SINGAM)                      STATEF
      SINPSI = SIN(PSI)                                                 STATEF
      COSPSI = COS(PSI)                                                 STATEF
      SINRHO = SIN(RHO)                                                 STATEF
      COSRHO = COS(RHO)                                                 JUL19B
      SIN2RO = SINRHO*COSRHO                                            STATEF
```

```
      SUBROUTINE STATEF                                                 STATEF
C                                                                       STATEF
C                   THIS ROUTINE COMPUTES ALL DYNAMIC QUANTITIES WHICH  STATEF
C                   BEAR NO EXPLICIT DEPENDENCE ON THE CONTROL          STATEF
C                                                                       STATEF
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER     GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI     , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX     , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA     , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG     , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB     , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT       , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE      ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMA, ISPAT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION DYN(1)                                                  STATEF
      EQUIVALENCE (DYN, XX)                                             STATEF
      DATA XSTATF/6HSTATEF/                                             STATEF
      DATA DEG/57.2957795130823/                                        STATEF
      SINGAM = SIN(GAM)                                                 STATEF
      COSGAM = COS(GAM)                                                 JUL19B
      COS2GM = (COSGAM + SINGAM)*(COSGAM - SINGAM)                      STATEF
      SINPSI = SIN(PSI)                                                 STATEF
      COSPSI = COS(PSI)                                                 STATEF
      SINRHO = SIN(RHO)                                                 STATEF
      COSRHO = COS(RHO)                                                 JUL19B
      SIN2RO = SINRHO*COSRHO                                            STATEF
```

```
          SIN2RO = SIN2RO + SIN2RO                                       STATEF
          COS2RO = (COSRHO + SINRHO)*(COSRHO - SINRHO)                   STATEF
          R = ALT + ER                                                   JUL21
C                    GRAVITY AND ITS PARTIALS.                           STATEF
          G = GM/R**2                                                    STATEF
          GH = -G/R                                                      STATEF
          GH = GH + GH                                                   STATEF
          GRR = -GH/R                                                    STATEF
          GRR = GRR + GRR + GRR                                          STATEF
          W = GR*M                                                       STATEF
          XX = X + 1. - XARC                                             STATEF
          TIME = TSTART + XX*TAU                                         STATEF
          TIMES = TIME - TSTAGE                                          STATEF
C                    IS THIS FREE FALL.                                  STATEF
          IF(KODE .EQ. 0) RETURN                                         STATEF
C                    NO.  COMPUTE ATMOSPHERIC AND/OR THRUST AFFECTS      STATEF
          IF(IATM - 1) 101, 102, 103                                     STATEF
C                    *62 STANDARD ATMOSPHERE                             STATEF
  101 CALL ANL625(ALT, TAMP, RORRR, IDAM)                                STATEF
          GO TO 104                                                      STATEF
C                    *63 PATRICK AFB ATMOSPHERE                          STATEF
  102 CALL ANL63P(ALT, TAMP, RORRR, IDAM)                                STATEF
          GO TO 104                                                      STATEF
C                    VACUUM.  CHECK FOR CG TRACK.                        STATEF
  103 IF(JAER - 2) 108, 108, 107                                         STATEF
C                    COMP. MACH AND DYNAMIC PRESSURE                     STATEF
  104 MACH = V/CS                                                        STATEF
          FCTR = V*V/2.                                                  STATEF
          Q = FCTR*RO                                                    STATEF
C                    IS REFERENCE AREA POSITIVE                          STATEF
          IF(SREF .LE. 0.) GO TO 108                                     STATEF
C                    YES.  COMPUTE PARTIALS OF MACH AND DYN. PRESSURE    STATEF
          MACHV = 1./CS                                                  STATEF
          MACHR = -MACH*MACHV*CSR                                        STATEF
          MACHVR = MACHR/V                                               STATEF
          MACHRR = MACHR*CSR                                             STATEF
          MACHRR = -MACHV*(MACHRR + MACHRR + MACH*CSRR)                  STATEF
          QV = V*RO                                                      STATEF
          QR = FCTR*ROR                                                  STATEF
          QVV = RO                                                       STATEF
          QVR = V*ROR                                                    STATEF
          QRR = FCTR*RORR                                                STATEF
C                    IS THIS BIVARIANT AERO.                             STATEF
          IF(JAER .EQ. 2) GO TO 108                                      STATEF
C                    NO.  UNIVARIANT. COMPUTE COEFFICIENTS NEEDED FOR    STATEF
C                    LINEAR AERO MODEL.                                  STATEF
          CALL SPLINE(MAEA, MACH, CLA, CLAM, CLAMM)                      STATEF
          CALL SPLINE(MAEB, MACH, CLO, CLOM, CLOMM)                      STATEF
          CALL SPLINE(MAEC, MACH, CDO, CDOM, CDOMM)                      STATEF
          CALL SPLINE(MAED, MACH,  FK,  FKM,  FKMM)                      STATEF
          CLA = CLA*DEG                                                  STATEF
          CLAM = CLAM*DEG                                                STATEF
          CLAMM = CLAMM*DEG                                              STATEF
C                    IS MOMENT BALANCING CALLED FOR.                     STATEF
          IF(JAER .EQ. 1) GO TO 108                                      STATEF
C                    YES.  CHECK TO SEE IF OUST SIMPLE CG TRACK          STATEF
          IF(DREF .LE. 0.) GO TO 105                                     STATEF
C                    NO.  UNTRIM AERO DATA.  COMPUTE COEFF. NEEDED FOR   STATEF
C                    MOMENT BALANCING                                    STATEF
          CALL SPLINE(MAEE, MACH, CMO, CMOM, CMOMM)                      STATEF
          CALL SPLINE(MAEF, MACH, CMA, CMAM, CMAMM)                      STATEF
          CMA = CMA*DEG                                                  STATEF
          CMAM = CMAM*DEG                                                STATEF
          CMAMM = CMAMM*DEG                                              STATEF
C                    IS THIS POWERED FLIGHT                              STATEF
  105 IF(IPOW .EQ. 0) GO TO 107                                          STATEF
C                    YES.  COMPUTE BLEND FACTOR AND PARTIALS.            STATEF
  106 CALL SPLINE(MAEG, Q, XJ, XJQ, XJQQ)                                STATEF
          IF(XJ .LT. 0. .OR. 1. .LT. XJ) CALL ERROR(XSTATF, -1, 1)       STATEF
          XJV = XJQ*QV                                                   STATEF
          XJR = XJQ*QR                                                   STATEF
          XJVV = XJQ*QVV + XJQQ*QV*QV                                    STATEF
          XJVR = XJQ*QVR + XJQQ*QV*QR                                    STATEF
```

```
      XJRR = XJQ*QRR + XJQQ*QR*QR                                   STATEF
C                 COMPUTE CG COORDS. AND THEIR PARTIALS             STATEF
  107 CALL SPLINE(MXCG, W, XCG, XCGM, XCGMM)                        STATEF
      CALL SPLINE(MZCG, W, ZCG, ZCGM, ZCGMM)                        STATEF
      XCGM = XCGM*GR                                                STATEF
      ZCGM = ZCGM*GR                                                STATEF
      XCGMM = XCGMM*GR*GR                                           STATEF
      ZCGMM = ZCGMM*GR*GR                                           STATEF
C                 IS THIS POWERED FLIGHT                            STATEF
  108 IF(IPOW .EQ. 0) RETURN                                        STATEF
C                 YES.  IS VACUUM THRUST TABULAR.                   STATEF
C                 YES.  ARE WE POWERED BY AIRBREATHER               JUL21
      IF(J1 .NE. 4) GO TO 1081                                      JUL21
C                 AIRBREATHER.  COMPUTE THRUST AND SPEC. FUEL CONSUMP. AUG09
      CALL BLINE(V, ALT, TAIRB)                                     JUL21
      GO TO 1082                                                    JUL21
C                 NOT AIRBREATHER.  IS VAC. THRUST TABULAR.         JUL21
 1081 IF(MTT .LE. 0) RETURN                                         JUL21
      IF(MTT .LE. 0) RETURN                                         STATEF
C                 YES.  COMPUTE NOMINAL, I.E. UNTHROTTLED, VACUUM   STATEF
C                 THRUST                                            STATEF
      CALL SPLINE(MTT, TIMES, FVAC, FVACT, FVACTT)                  STATEF
      IF(FVAC .LT. 0.) CALL ERROR(XSTATF, -2, 1)                    STATEF
      FVAC = TMULT*FVAC                                             STATEF
      FVACT = TMULT*FVACT*XX                                        STATEF
      FVACTT = TMULT*FVACTT*XX**2                                   STATEF
C                 COMPUTE BASE DRAG IF INPUT.                       STATEF
 1082 IF(IPOW .EQ. 2) CALL SPLINE(MDB, ALT, DB, DBR, DBRR)          JUL21
      RETURN                                                        STATEF
      END                                                           STATEF
```

12449

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALT	h	I	Altitude	(FT)	/D	/(94)	OUTPUT	I	ALT
							STATEF	I	ALT
							WRAPUP	I	ALT
CDO	C_{D_0}	I	Drag coefficient at $\alpha = 0$		/DYNA	/(104)	AEROCO	I	CDO
							STATEF	I	CDO
CDOM	$\partial C_{D_0}/\partial M$	I	See symbol		/DYNA	/(105)	AEROCO	I	CDOM
							STATEF	I	CDOM
CDOMM	$\partial^2 C_{D_0}/\partial M^2$	I	See symbol		/DYNA	/(145)	AEROCO	I	CDOMM
							STATEF	I	CDOMM
CLA	C_{L_α}	M	Lift coefficient slope	(RAD^{-1})	/DYNA	/(187)	AEROCO	M	CLA
							STATEF	M	CLA
							UT	I	CLA
CLAM	$\partial C_{L_\alpha}/\partial M$	M	See symbol		/DYNA	/(191)	AEROCO	M	CLAM
							STATEF	M	CLAM
							UT	I	CLAM
CLAMM	$\partial^2 C_{L_\alpha}/\partial M^2$	M	See symbol		/DYNA	/(146)	AEROCO	I	CLAMM
							STATEF	M	CLAMM
CLO	C_{L_0}	I	Lift coefficient at $\alpha = 0$		/DYNA	/(106)	AEROCO	I	CLO
							STATEF	I	CLO
CLOM	$\partial C_{L_0}/\partial M$	I	See symbol		/DYNA	/(147)	AEROCO	I	CLOM
							STATEF	I	CLOM
CLOMM	$\partial^2 C_{L_0}/\partial M^2$	I	See symbol		/DYNA	/(148)	AEROCO	I	CLOMM
							STATEF	I	CLOMM
CMA	C_{m_α}	M	Moment coefficient slope	(RAD^{-1})	/DYNA	/(123)	MOMECO	I	CMA
							STATEF	M	CMA
							UT	I	CMA
CMAM	$\partial C_{m_\alpha}/\partial M$	M	See symbol		/DYNA	/(127)	MOMECO	I	CMAM
							STATEF	M	CMAM
							UT	I	CMAM
CMAMM	$\partial^2 C_{m_\alpha}/\partial M^2$	M	See symbol		/DYNA	/(131)	MOMECO	I	CMAMM
							STATEF	M	CMAMM
CMO	C_{m_0}	I	Moment coefficient at $\alpha = 0$.		/DYNA	/(128)	MOMECO	I	CMO
							STATEF	I	CMO
CMOM	$\partial C_{m_0}/\partial M$	I	See symbol		/DYNA	/(129)	MOMECO	I	CMOM
							STATEF	I	CMOM
CMOMM	$\partial^2 C_{m_0}/\partial M^2$	I	See symbol		/DYNA	/(130)	MOMECO	I	CMOMM
							STATEF	I	CMOMM
COSGAM	$\cos\gamma$	M	See symbol		/DYNA	/(4)	AL1	I	COSGAM
							AL4	I	COSGAM
							AL7	I	COSGAM
							AL8	I	COSGAM
							AL9	I	COSGAM
							CONTRL	I	COSGAM
							NLDRV	I	COSGAM
							OUTPUT	I	COSGAM
							PDBCOL	I	COSGAM
							STATEF	M	COSGAM
COSPSI	$\cos\psi$	O	See symbol		/DYNA	/(95)	AL4	I	COSPSI
							AL7	I	COSPSI
							AL8	I	COSPSI
							AL9	I	COSPSI
							CONTRL	I	COSPSI
							NLDRV	I	COSPSI
							PDBCOL	I	COSPSI
							STATEF	O	COSPSI
COSRHO	$\cos\rho$	M	See symbol		/DYNA	/(97)	AL4	I	COSRHO
							AL7	I	COSRHO
							AL8	I	COSRHO
							AL9	I	COSRHO
							CONTRL	I	COSRHO
							NLDRV	I	COSRHO
							OUTPUT	I	COSRHO
							PDBCOL	I	COSRHO
							STATEF	M	COSRHO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
COS2GM	$\cos 2\gamma$	O	See symbol	/DYNA /(	121)	AL4 STATEF	I O	COS2GM COS2GM
COS2RO	$\cos 2\rho$	O	See symbol	/DYNA /(	120)	AL4 AL7 AL8 NLDRV STATEF	I I I I O	COS2RO COS2RO COS2RO COS2RO COS2RO
CS	a	I	Speed of sound (FT/SEC)	/DYNA /(	16)	OUTPUT STATEF	I I	CS CS
CSR	$\partial a/\partial R$	I	See symbol	/DYNA /(	20)	STATEF	I	CSR
CSRR	$\partial^2 a/\partial R^2$	I	See symbol	/DYNA /(	24)	STATEF	I	CSRR
DB	D_b	I	Base drag (LBS)	/DYNA /(	163)	AL1 AL4 AL6 AL7 AL8 AL9 APPLY CONTRL NLDRV OUTPUT STATEF TH3 UT	I I I I I I I I I I I I I	DB DB DB DB DB DB DB DB DB DB DB DB DB
DBR	$\partial D_b/\partial R$	I	See symbol	/DYNA /(	86)	AL1 AL4 AL6 AL7 AL8 AL9 APPLY STATEF TH3 UT	I I I I I I I I I I	DBR DBR DBR DBR DBR DBR DBR DBR DBR DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol	/DYNA /(	87)	AL4 AL6 AL7 AL8 AL9 APPLY STATEF TH3 UT	I I I I I I I I I	DBRR DBRR DBRR DBRR DBRR DBRR DBRR DBRR DBRR
DREF	D_{ref}	I	Aerodynamic reference length	/ARCDAT/(	37)	STATEF UT	I I	DREF DREF
ER	E_R	I	Earth radius. (FT)	/GLOBAL/(	2)	ENVPRO PDBCOL QLTOSZ STATEF	I I I I	ER ER ER ER
FK	k	I	Induced drag coefficient	/DYNA /(	107)	AEROCO STATEF	I I	FK FK
FKM	$\partial k/\partial M$	I	See symbol	/DYNA /(	182)	AEROCO STATEF	I I	FKM FKM
FKMM	$\partial^2 k/\partial M^2$	I	See symbol	/DYNA /(	183)	AEROCO STATEF	I I	FKMM FKMM
FVAC		M	Total vacuum thrust (rocket) (LBS)	/DYNA /(	33)	APPLY ARCIN IMPULS NLDRV STATEF TH2	I M M I M I	FVAC FVAC FVAC FVAC FVAC FVAC
FVACT		M	Not used.	/DYNA /(	37)	ARCIN STATEF TH2	I M I	FVACT FVACT FVACT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
FVACTT		M	Not used.	/DYNA /(	41)	ARCIN	I	FVACTT
						STATEF	M	FVACTT
						TH2	I	FVACTT
G	g	M	Instantaneous gravitational acceleration (FT/SEC^2)	/DYNA /(	8)	AL4	I	G
						AL7	I	G
						AL8	I	G
						AL9	I	G
						CONTRL	I	G
						NLDRV	I	G
						STATEF	M	G
GAM	γ	I	Relative flight path angle. (RAD)	/D /(	92)	ARCIN	I	GAM
						ENVPRO	I	GAM
						OUTPUT	I	GAM
						STATEF	I	GAM
						WRAPUP	I	GAM
GH	$\partial g/\partial R$	M	See symbol	/DYNA /(	142)	AL7	I	GH
						AL8	I	GH
						NLDRV	I	GH
						STATEF	M	GH
GM	GM	I	Product of Newton's universal gravitational constant and the mass of the earth. (FT^3/SEC^2)	/GLOBAL/(	67)	OUTPUT	I	GM
						PDBCOL	I	GM
						STATEF	I	GM
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCOL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
GRR	$\partial^2 g/\partial R^2$	M	See symbol	/DYNA /(	143)	AL7	I	GRR
						AL8	I	GRR
						NLDRV	I	GRR
						STATEF	M	GRR
IATM		I	Atmosphere option flag	/ARCDAT/(	7)	ARCIN	I	IATM
						NLDRV	I	IATM
						OUTPUT	I	IATM
						STATEF	I	IATM
IDAM		I	Optional atmospheric calculations flag. IDAM = -1: Compute $\partial^3\rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3\rho_a/\partial R^3$, ρ_a, $\partial\rho_a/\partial R$, etc.	/DYNA /(	218)	ARCIN	O	IDAM
						ERROR	I	IDAM
						NPLANE	O	IDAM
						STATEF	I	IDAM
						WRAPUP	O	IDAM
IPOW		I	Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag	/DYNA /(	139)	ARCIN	M	IPOW
						FORCES	I	IPOW
						NPLANE	I	IPOW
						STATEF	I	IPOW
						THROTL	I	IPOW
JAER		I	Aerodynamic model option flag	/ARCDAT/(	9)	AEROCO	I	JAER
						ARCIN	I	JAER
						OUTPUT	I	JAER
						STATEF	I	JAER
						UT	I	JAER
J1		I	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY	I	J1
						ARCIN	O	J1
						CONTRL	M	J1
						FORCES	I	J1
						NPLANE	I	J1
						STATEF	I	J1
						THROTL	M	J1

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
KODE		I	Steering vector flag KODE = 0: Free fall, $\alpha = \phi = 0$; KODE = 1: Both α and ϕ optimal ; KODE = 2: α optimal and $\phi = 0$; KODE = 3: α nonoptimal and ϕ optimal; KODE = 4: Vertical rise or pitchover; KODE = 5: α nonoptimal and $\phi = 0$.	/DYNA /(	25)	APPLY	I	KODE
						ARCIN	O	KODE
						CONTRL	M	KODE
						FORCES	I	KODE
						NLDRV	I	KODE
						STATEF	I	KODE
M	m	I	Mass (G'S)	/D /(	97)	AL4	I	M
						AL7	I	M
						AL8	I	M
						AL9	I	M
						APPLY	I	M
						BRANPT	I	M
						COSTAB	I	M
						COSTAI	I	M
						INTRPT	I	M
						NLDRV	I	M
						OUTPUT	I	M
						SALVE	I	M
						STATEF	I	M
						WRAPUP	I	M
MACH	M	M	Mach number	/DYNA /(	26)	AEROCO	I	MACH
						ENVPRQ	I	MACH
						OUTPUT	I	MACH
						STATEF	M	MACH
MACHR	$\partial M/\partial R$	M	See symbol	/DYNA /(	44)	STATEF	M	MACHR
						UT	I	MACHR
MACHRR	$\partial^2 M/\partial R^2$	M	See symbol	/DYNA /(	118)	STATEF	M	MACHRR
						UT	I	MACHRR
MACHV	$\partial M/\partial V$	M	See symbol	/DYNA /(	43)	STATEF	M	MACHV
						UT	I	MACHV
MACHVR	$\partial^2 M/\partial V \partial R$	O	See symbol	/DYNA /(	117)	STATEF	O	MACHVR
						UT	I	MACHVR
MAEA		I	Curve number	/ARCDAT/(	18)	STATEF	I	MAEA
MAEB		I	Curve number	/ARCDAT/(	19)	STATEF	I	MAEB
MAEC		I	Curve number	/ARCDAT/(	20)	STATEF	I	MAEC
MAED		I	Curve number	/ARCDAT/(	21)	STATEF	I	MAED
MAEE		I	Curve number	/ARCDAT/(	22)	STATEF	I	MAEE
MAEF		I	Curve number	/ARCDAT/(	23)	STATEF	I	MAEF
MAEG		I	Curve number	/ARCDAT/(	24)	STATEF	I	MAEG
MDB		I	Curve number - base drag table	/ARCDAT/(	31)	ARCIN	I	MDB
						STATEF	I	MDB
MTT		I	Table number for tabulated rocket vacuum thrust	/DYNA /(	172)	ARCIN	M	MTT
						STATEF	I	MTT
MXCG		I	Curve number -xcg table	/ARCDAT/(	27)	STATEF	I	MXCG
MZCG		I	Curve number- zcg table	/ARCDAT/(	28)	STATEF	I	MZCG
PSI	ψ	I	Relative azimuth angle. (RAD)	/D /(	93)	OUTPUT	I	PSI
						STATEF	I	PSI
						WRAPUP	I	PSI
Q	q	M	Dynamic pressure (LBS/FT^2)	/DYNA /(	27)	ENVPRQ	I	Q
						OUTPUT	I	Q
						PDBCOL	I	Q
						STATEF	M	Q
						UT	I	Q
QR	$\partial q/\partial R$	M	See symbol	/DYNA /(	29)	PDBCOL	I	QR
						STATEF	M	QR
						UT	I	QR
QRR	$\partial^2 q/\partial R^2$	M	See symbol	/DYNA /(	32)	STATEF	M	QRR
						UT	I	QRR
QV	$\partial q/\partial V$	M	See symbol	/DYNA /(	28)	PDBCOL	I	QV
						STATEF	M	QV
						UT	I	QV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
QVR	$\partial^2 q/\partial V \partial R$	M	See symbol	/DYNA	/(31)	STATEF	M	QVR
						UT	I	QVR
QVV	$\partial^2 q/\partial V^2$	M	See symbol	/DYNA	/(30)	STATEF	M	QVV
						UT	I	QVV
R	R	M	Radial distance from earth center to vehicle (FT)	/DYNA	/(7)	AL4	I	R
						AL7	I	R
						AL8	I	R
						AL9	I	R
						CONTRL	I	R
						ENVPRO	I	R
						NLDRV	I	R
						PDBCOL	I	R
						QLTOSZ	I	R
						STATEF	M	R
RHO	ρ	I	Latitude (RAD)	/D	/(95)	AL9	I	RHO
						OUTPUT	I	RHO
						STATEF	I	RHO
						WRAPUP	I	RHO
RO	ρ_a	I	Atmospheric density (SLGS/FT^3)	/DYNA	/(15)	AL7	I	RO
						AL8	I	RO
						AL9	I	RO
						NLDRV	I	RO
						OUTPUT	I	RO
						PDBCOL	I	RO
						STATEF	I	RO
ROR	$\partial \rho_a/\partial R$	I	See symbol	/DYNA	/(19)	AL7	I	ROR
						AL8	I	ROR
						AL9	I	ROR
						NLDRV	I	ROR
						PDBCOL	I	ROR
						STATEF	I	ROR
RORR	$\partial^2 \rho_a/\partial R^2$	I	See symbol	/DYNA	/(23)	AL7	I	RORR
						AL8	I	RORR
						AL9	I	RORR
						NLDRV	I	RORR
						STATEF	I	RORR
RORRR	$\partial^3 \rho_a/\partial R^3$	I	See symbol	/DYNA	/(213)	AL7	I	RORRR
						AL8	I	RORRR
						AL9	I	RORRR
						STATEF	I	RORRR
SINGAM	$\sin\gamma$	M	See symbol	/DYNA	/(3)	AL1	I	SINGAM
						AL4	I	SINGAM
						AL7	I	SINGAM
						AL8	I	SINGAM
						AL9	I	SINGAM
						CONTRL	I	SINGAM
						NLDRV	I	SINGAM
						PDBCOL	I	SINGAM
						STATEF	M	SINGAM
SINPSI	$\sin\psi$	O	See symbol	/DYNA	/(94)	AL4	I	SINPSI
						AL7	I	SINPSI
						AL8	I	SINPSI
						AL9	I	SINPSI
						CONTRL	I	SINPSI
						NLDRV	I	SINPSI
						PDBCOL	I	SINPSI
						STATEF	O	SINPSI
SINRHO	$\sin\rho$	M	See symbol	/DYNA	/(96)	AL4	I	SINRHO
						AL7	I	SINRHO
						AL8	I	SINRHO
						AL9	I	SINRHO
						CONTRL	I	SINRHO
						NLDRV	I	SINRHO
						OUTPUT	I	SINRHO
						PDBCOL	I	SINRHO
						STATEF	M	SINRHO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SIN2RO	$\sin 2\rho$	M	See symbol	/DYNA /(	119)	AL4	I	SIN2R
						AL7	I	SIN2R
						AL8	I	SIN2R
						NLDRV	I	SIN2R
						STATEF	M	SIN2R
SREF	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
						BNDRY	I	ARCDA
						CHECK	I	ARCDA
						FETCH	I	ARCDA
						SALVE	I	ARCDA
						STATEF	I	SREF
						UT	I	SREF
						WRAPUP	I	ARCDA
TAIRB		I	Air-breather engine thrust (LBS)	/DYNA /(	219)	STATEF	I	TAIRB
						TH4	I	TAIRB
TAMP	T_a	I	Atmospheric temperature (DEG-R)	/DYNA /(	13)	STATEF	I	TAMP
TAU	τ	I	Subarc duration (SEC)	/D /(	98)	ARCEN	I	TAU
						INARC	M	TAU
						NLDRV	I	TAU
						OUTPUT	I	TAU
						STATEF	I	TAU
TIME		M	Trajectory time (SEC)	/DYNA /(	2)	ENVPRO	I	TIME
						OUTPUT	I	TIME
						PDBCOL	I	TIME
						STATEF	M	TIME
						WRAPUP	I	TIME
TIMES		M	Elapsed burning time of present rocket engine (SECS)	/DYNA /(	168)	STATEF	M	TIMES
TMULT	T_{mult}	I	Thrust multiplier or number of engines	/ARCDAT/(	9)	ARCIN	I	TMULT
						STATEF	I	TMULT
TSTAGE		I	Trajectory time at which present rocket engine ignited. (SECS)	/DYNA /(	167)	ARCIN	M	TSTAGE
						STATEF	I	TSTAGE
						TRAJIN	O	TSTAGE
TSTART		I	Trajectory time at which present subarc commenced.	/DYNA /(	141)	ARCEN	M	TSTART
						ARCIN	M	TSTART
						STATEF	I	TSTART
						TRAJIN	O	TSTART
V	V	I	Relative velocity. (FT/SEC)	/D /(	91)	AL1	I	V
						AL4	I	V
						AL7	I	V
						AL8	I	V
						AL9	I	V
						BCOND	I	NOM
						BNDRY	O	NOM
						BRANPT	M	NOM
						CONTRL	I	V
						ENDPT	I	NOM
						ENVPRO	I	V
						FETCH	O	NOM
						INTERP	M	V
						INTRPT	M	NOM
						NLDRV	O	NOM
						NLDRV	I	V
						OUTPUT	I	V
						PDBCOL	I	V
						STATEF	I	V
						WRAPUP	I	V
W	W	M	Weight (LBS)	/DYNA /(	91)	AL5	I	W
						ENVPRO	I	W
						OUTPUT	I	W
						PDBCOL	I	W
						QLTOSZ	I	W
						STATEF	M	W
						TH3	I	W

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
X	x	I	The quasitime variable.		/D	/(1)	AL4	I	X
							BNDRY	O	X
							ERROR	I	X
							FETCH	O	X
							FORCES	I	X
							INARC	M	X
							INTERP	I	X
							MADAMS	M	X
							RKUTT1	M	X
							RKUTT2	M	X
							SALVE	M	X
							STATEF	I	X
							WRAPUP	M	TT
XARC		I	Quasitime at which present subarc commenced.		/DYNA	/(140)	ARCIN	O	XARC
							STATEF	I	XARC
XCG	X_{CG}	I	Center of gravity body x station	(FT)	/DYNA	/(157)	DL2	I	XCG
							STATEF	I	XCG
							UT	I	XCG
XCGM	$\partial X_{CG}/\partial m$	M	See symbol		/DYNA	/(108)	DL2	I	XCGM
							STATEF	M	XCGM
							UT	I	XCGM
XCGMM	$\partial^2 X_{CG}/\partial m^2$	M	See symbol		/DYNA	/(109)	DL2	I	XCGMM
							STATEF	M	XCGMM
							UT	I	XCGMM
XJ	$\dot{j}$	I	Control blend factor		/DYNA	/(159)	ARCIN	O	XJ
							DL2	I	XJ
							OUTPUT	I	XJ
							STATEF	I	XJ
							UT	I	XJ
XJR	$\partial \dot{j}/\partial R$	O	See symbol		/DYNA	/(113)	DL2	I	XJR
							STATEF	O	XJR
							UT	I	XJR
XJRR	$\partial^2 \dot{j}/\partial R^2$	O	See symbol		/DYNA	/(116)	DL2	I	XJRR
							STATEF	O	XJRR
							UT	I	XJRR
XJV	$\partial \dot{j}/\partial V$	O	See symbol		/DYNA	/(112)	DL2	I	XJV
							STATEF	O	XJV
							UT	I	XJV
XJVR	$\partial^2 \dot{j}/\partial V \partial R$	O	See symbol		/DYNA	/(115)	DL2	I	XJVR
							STATEF	O	XJVR
							UT	I	XJVR
XJVV	$\partial^2 \dot{j}/\partial V^2$	O	See symbol		/DYNA	/(114)	DL2	I	XJVV
							STATEF	O	XJVV
							UT	I	XJVV
XX		M	Fraction of subarc that has transpired		/DYNA	/(1)	ARCIN	O	XX
							ERROR	I	XX
							OUTPUT	I	XX
							STATEF	M	XX
ZCG	Z_{CG}	I	Center of gravity body z station	(FT)	/DYNA	/(158)	DL2	I	ZCG
							STATEF	I	ZCG
							UT	I	ZCG
ZCGM	$\partial Z_{CG}/\partial m$	M	See symbol		/DYNA	/(110)	DL2	I	ZCGM
							STATEF	M	ZCGM
							UT	I	ZCGM
ZCGMM	$\partial^2 Z_{CG}/\partial m^2$	M	See symbol		/DYNA	/(111)	DL2	I	ZCGMM
							STATEF	M	ZCGMM
							UT	I	ZCGMM

SUBRØUTINE THRØTL

Purpose

THROTL determines whether the rocket thrust should be throttled in order to satisfy a total acceleration limit.

```
      SUBROUTINE THROTL                                                 THROTL
C                                                                       THROTL
C                  THIS ROUTINE DETERMINES WHETHER THE THRUST SHOULD    THROTL
C                  BE THROTTLED IN ORDER TO SATISFY A POWERED ACCEL.    THROTL
C                  LIMIT.                                               THROTL
C                                                                       THROTL
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      ,ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      ,ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      ,ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      ,ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      ,ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        ,ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE      ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX     ,TIME   ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  , JUL21
     *COSA   ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   , DYNA
     *ROR    ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     , DYNA
     *QV     ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM , DYNA
     *FVACT  ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   , DYNA
     *ISPV   ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR , DYNA
     *ISPRM  ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA , DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA , DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM , DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,QE    ,TAX   , DYNA
     *W      ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR, DYNA
     *MUR    ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  , DYNA
     *XCGMM  ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR, DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  , DYNA
     *CMAM   ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR, DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA, DYNA
     *CDOMM  ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   , DYNA
     *SID    ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX , DYNA
     *DB     ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED  DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT    ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF , DYNA
     *ILOAD  ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  , DYNA
     *CLMM   ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198, DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA, DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216, DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   , JUL21
     *SFCV   ,SFCH  ,SFCVV ,SFCHH ,SFCVH                               AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1     ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  , MATS
     *XK1D   ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   , MATS
     *XM19   ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD , MATS
     *XK3TD  ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA , MATS
     *XK3DA  ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  , MATS
     *XK1V   ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  , MATS
     *XK1R   ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  , MATS
     *XK1M   ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT , MATS
     *XK1VD  ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT , MATS
     *XK1GD  ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT , MATS
     *XK1PD  ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT , MATS
     *XK1RD  ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT , MATS
     *XK1OD  ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT , MATS
     *XK1UD  ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT   MATS
      COMMON /MATS/                                                     MATS
     *XK1MD  ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT , MATS
     *XK1ZD  ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV , MATS
     *XK1GV  ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV , MATS
     *XK1OV  ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV , MATS
     *XK1ZV  ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG , MATS
     *XK1RG  ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG , MATS
     *XK1MG  ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP , MATS
```

```
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,     MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,     MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,     MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,     MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,     MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,     MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,     MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2               MATS
      COMMON /MATS/                                                      MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                      MATS
     *PV    ,PG    ,PP    ,PR    ,PD    ,PVV   ,PGV   ,PPV   ,PRV   ,   MATS
     *PDV   ,PGG   ,PPG   ,PRG   ,PDG   ,PPP   ,PRP   ,PDP   ,PRR   ,   MATS
     *PDR   ,PDD   ,PLG   ,PLP                                          MATS
      EQUIVALENCE(PROD1,PROD5)                                          MATS
      SWITCH = .FALSE.                                                  THROTL
      IF(GMAX .LE. 0 .OR. IPOW .EQ. 0 .OR. J1 .GT. 4) RETURN            JUL21
      CALL TH3000                                                       THROTL
      IF(XK1 .LE. 0.) RETURN                                            THROTL
      J1 = 3                                                            THROTL
      SWITCH = .TRUE.                                                   THROTL
      RETURN                                                            THROTL
      END                                                               THROTL
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GMAX	G_{MAX}	I	Maximum total acceleration g load	/ARCDAT/(	12)	AL5 NPLANE THROTL TH3	I I I I	GMAX GMAX GMAX GMAX
IPOW		I	Powered flag. IPOW = 0: No thrust and no base drag IPOW = 1: Thrust, but no base drag IPOW = 2: Thrust and base drag	/DYNA /(	139)	ARCIN FORCES NPLANE STATEF THROTL	M I I I I	IPOW IPOW IPOW IPOW IPOW
J1		M	Thrust option flag. J1 = 1: Constant thrust; J1 = 2: Input vacuum thrust; J1 = 3: Powered total acceleration limit; J1 = 4: Air-breather engine.	/DYNA /(	173)	APPLY ARCIN CONTRL FORCES NPLANE STATEF THROTL	I O M I I I M	J1 J1 J1 J1 J1 J1 J1
SWITCH		O	Logical flag that is true if this is the compute point at which the powered acceleration constraint commences.	/DYNA /(	184)	CONTRL NPLANE THROTL	I I O	SWITCH SWITCH SWITCH
XK1		I	First entry of 3 word in-plane control constraints K	/MATS /(	4)	ALGCON THROTL TH1 TH2 TH3 TH4	I I O O O O	XK1 XK1 XK1 XK1 XK1 XK1

SUBRØUTINE
TH1

Purpose

T111 evaluates the constant thrust constraint

$$T - C_T = 0.$$

```
      SUBROUTINE TH1                                                    TH1
C                                                                       TH1
C                    THIS ROUTINE APPLIES WHEN NET THRUST IS A CONSTANT TH1
C                                                                       TH1
C                                                                       TH1
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMA, ISPAT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MAGH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPAT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTAM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GN    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CDBAE ,SIDAE ,CDD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEV(2, 8), DPOL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
```

1264

```
     *PV     ,PG     ,PP     ,PR     ,PO     ,PVV    ,PGV    ,PPV    ,PRV    ,       MATS
     *POV    ,PGG    ,PPG    ,PRG    ,POG    ,PPP    ,PRP    ,POP    ,PRR    ,       MATS
     *POR    ,POO    ,PLG    ,PLP                                                    MATS
      EQUIVALENCE(PROD1,PROD5)                                                       MATS
C                                                                                    TH1
      ENTRY TH1002                                                                   TH1
      ENTRY TH1001                                                                   TH1
   40 XK1T = 1.                                                                      TH1
      ENTRY TH1000                                                                   TH1
   50 XK1 = T - CT                                                                   TH1
C                                                                                    TH1
      RETURN                                                                         TH1
      END                                                                            TH1
```

1265

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
CT	C_T	I	Value for thrust in case constant thrust constraint is used. (LBS)	/DYNA	/(150)	TH1	I	CT
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
XK1		O	First entry of 3 word in-plane control constraints K	/MATS	/(4)	ALGCON	I	XK1
						THROTL	I	XK1
						TH1	O	XK1
						TH2	O	XK1
						TH3	O	XK1
						TH4	O	XK1
XK1T	$K_T^{(1)}$	O	First entry of 3x3 matrix containing the explicit partials of K with respect to ω, K_ω	/MATS	/(7)	ALGCON	I	XK1T
						TH1	O	XK1T
						TH2	O	XK1T
						TH3	M	XK1T

SUBRØUTINE
TH2

Purpose

TI12 evaluates the unthrottled vacuum thrust constraint

$$T - F_{VAC} + A_E P_a = 0$$

In addition, it evaluates the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE TH2                                                     TH2
C                                                                        TH2
C                 THIS ROUTINE APPLIES WHEN THRUST IS NOT THROTTLED      TH2
C                                                                        TH2
C                                                                        TH2
      LOGICAL SWITCH, ILOAD                                              DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,       DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,      DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                                DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                  DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                         DYNA
      COMMON /DYNA/                                                      DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,    JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,    DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,    DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,    DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,    DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,    DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,    DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,    DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,    DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,    DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,    DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,    DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,    DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,    DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,    DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,    DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,    DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,    DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED     DYNA
      COMMON /DYNA/                                                      DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,    DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,    DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,    DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,    DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,    DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,    JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                  AUG09
      DIMENSION PROD1(2, 64)                                             MATS
      COMMON /MATS/                                                      MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,    MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,    MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,    MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,    MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,    MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,    MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,    MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,    MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,    MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,    MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,    MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,    MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,    MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT      MATS
      COMMON /MATS/                                                      MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,    MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,    MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,    MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,    MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,    MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,    MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,    MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,    MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,    MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,    MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,    MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,    MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,    MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,    MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2               MATS
      COMMON /MATS/                                                      MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)   MATS
      COMMON /MATS/                                                      MATS
```

```
     *PV     ,PG     ,PP     ,PR     ,PO     ,PVV    ,PGV    ,PPV    ,PRV    ,      MATS
     *POV    ,PGG    ,PPG    ,PRG    ,POG    ,PPP    ,PBP    ,POP    ,PRR    ,      MATS
     *POR    ,POO    ,PLG    ,PLP                                                   MATS
      EQUIVALENCE(PROD1,PROD5)                                                      MATS
C                                                                                   TH2
C                      THIS ENTRY COMP. 2ND PARTS. W/RESP. TO STATE                 TH2
      ENTRY TH2020                                                                  TH2
      XK1RR = AE*PARR                                                               TH2
      XK1ZZ = -FVACTT                                                               TH2
C                      THESE ENTRIES COMP. 1ST PARTS. W/RESP. TO STATE              TH2
      ENTRY TH2011                                                                  TH2
      ENTRY TH2010                                                                  TH2
      XK1R = AE*PAR                                                                 TH2
      XK1Z = -FVACT                                                                 TH2
C                      THESE ENTRIES COMP. 1ST PARTS. W/RESP. TO CONTROL.           TH2
      ENTRY TH2002                                                                  TH2
      ENTRY TH2001                                                                  TH2
   40 XK1T = 1.                                                                     TH2
C                      THIS ENTRY COMP. THE CONSTRAINING EQ. ONLY                   TH2
      ENTRY TH2000                                                                  TH2
   50 XK1 = T - FVAC + AE*PA                                                        TH2
C                                                                                   TH2
      RETURN                                                                        TH2
      END                                                                           TH2
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AE	A_{exit}	I	Total nozzle exit area (FT^2)	/DYNA	/(89)	APPLY	I	AE
						ARCIN	O	AE
						IMPULS	I	AE
						NLDRV	I	AE
						TH2	I	AE
FVAC		I	Total vacuum thrust (rocket) (LBS)	/DYNA	/(33)	APPLY	I	FVAC
						ARCIN	M	FVAC
						IMPULS	M	FVAC
						NLDRV	I	FVAC
						STATEF	M	FVAC
						TH2	I	FVAC
FVACT		I	Not used.	/DYNA	/(37)	ARCIN	I	FVACT
						STATEF	M	FVACT
						TH2	I	FVACT
FVACTT		I	Not used.	/DYNA	/(41)	ARCIN	I	FVACTT
						STATEF	M	FVACTT
						TH2	I	FVACTT
PA	P_a	I	Atmospheric pressure (LBS/FT^2)	/DYNA	/(14)	IMPULS	I	PA
						NLDRV	I	PA
						OUTPUT	I	PA
						TH2	I	PA
PAR	$\partial P_a/\partial R$	I	See symbol	/DYNA	/(18)	APPLY	I	PAR
						TH2	I	PAR
PARR	$\partial^2 P_a/\partial R^2$	I	See symbol	/DYNA	/(22)	APPLY	I	PARR
						TH2	I	PARR
T	T	I	Thrust (LBS)	/DYNA	/(42)	ALGCON	M	T
						AL1	I	T
						AL4	I	T
						AL6	I	T
						AL7	I	T
						AL8	I	T
						AL9	I	T
						APPLY	I	T
						ARCIN	O	T
						CONTRL	M	T
						DL2	I	T
						IMPULS	I	T
						OUTPUT	I	T
						TH1	I	T
						TH2	I	T
						TH3	I	T
						TH4	I	T
XK1		O	First entry of 3 word in-plane control constraints K	/MATS	/(4)	ALGCON	I	XK1
						THROTL	I	XK1
						TH1	O	XK1
						TH2	O	XK1
						TH3	O	XK1
						TH4	O	XK1
XK1RR	$K_{hh}^{(1)}$	O	The first entry in a 3x5 matrix containing K_{yh}	/MATS	/(205)	TH2	O	XK1RR
						TH3	M	XK1RR
						TH4	O	XK1RR
XK1T	$K_T^{(1)}$	O	First entry of 3x3 matrix containing the explicit partials of K with respect to m, K_m	/MATS	/(7)	ALGCON	I	XK1T
						TH1	O	XK1T
						TH2	O	XK1T
						TH3	M	XK1T
XK1ZZ	$K_{yy}^{(1)}$	O	The first entry in a 3x1 matrix containing K_{yy}	/MATS	/(247)	TH2	O	XK1ZZ

SUBRØUTINE TH3

Purpose

TH3 evaluates the powered total acceleration constraint

$$(T\cos(\alpha-\delta_E) - D_B\cos\alpha - D)^2 + (T\sin(\alpha-\delta_E) - D_B\sin\alpha + L)^2 - (g_{max}W)^2 = 0$$

In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

```
      SUBROUTINE TH3                                                   TH3
C                                                                      TH3
C                THIS ROUTINE APPLIES WHEN THRUST IS THROTTLED         TH3
C                                                                      TH3
C                                                                      TH3
      COMMON/GLOBAL/                                                   GLOBAL
     *GR     ,ER     ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER , GLOBAL
     *ITRMAX,JJOP(6)       ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                         GLOBAL
      COMMON/ARCDAT/                                                   ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDD      , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                             ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                          ARCDAT
      LOGICAL SWITCH, ILOAD                                            DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,     DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMA, ISPMT, ISPTT, LIFT, LIFTV,    DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                              DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                       DYNA
      COMMON /DYNA/                                                    DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,  JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,  DYNA
     *ROR   ,CSR   ,TEMPRR,RARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,  DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,  DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,  DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,  DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,  DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,  DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,  DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,  DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,  DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,  DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,  DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,  DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,  DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,  DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,  DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,  DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED   DYNA
      COMMON /DYNA/                                                    DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,  DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,  DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,  DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,  DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,  DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,  JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                AUG09
      DIMENSION PROD1(2, 64)                                           MATS
      COMMON /MATS/                                                    MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,  MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,  MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,  MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,  MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,  MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,  MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,  MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,  MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,  MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,  MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,  MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,  MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,  MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT    MATS
      COMMON /MATS/                                                    MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,  MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,  MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,  MATS
```

```
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,    MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,    MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,    MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,    MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,    MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,    MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,    MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,    MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,    MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,    MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,    MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2               MATS
      COMMON /MATS/                                                      MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)   MATS
      COMMON /MATS/                                                      MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,    MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,    MATS
     *POR   ,POO   ,PLG   ,PLP                                           MATS
      EQUIVALENCE(PROD1,PROD5)                                           MATS
      ASF(X, Y) = T1*X + T2*Y                                            TH3
C                                                                        TH3
C                  THIS ENTRY COMP. 2ND PARTS.W/RESP.TO STATE            TH3
      ENTRY TH3020                                                       TH3
      ASSIGN 6 TO IGO                                                    TH3
      GO TO 4                                                            TH3
C                  THIS ENTRY COMP. MIXED PARTS.W/RESP.TO STATE AND      TH3
C                  COSTATE.                                              TH3
      ENTRY TH3011                                                       TH3
      ASSIGN 10 TO IGO                                                   TH3
      GO TO 4                                                            TH3
C                  THIS ENTRY COMP. 1ST PARTS.W/RESP.TO STATE            TH3
      ENTRY TH3010                                                       TH3
      ASSIGN 20 TO IGO                                                   TH3
      GO TO 4                                                            TH3
C                  THIS ENTRY COMP. 2ND PARTS.W/RESP.TO CONTROL          TH3
      ENTRY TH3002                                                       TH3
      ASSIGN 30 TO IGO                                                   TH3
      GO TO 5                                                            TH3
C                  THIS ENTRY COMP  1ST PARTS.W/RESP.TO CONTROL          TH3
      ENTRY TH3001                                                       TH3
      ASSIGN 40 TO IGO                                                   TH3
      GO TO 5                                                            TH3
C                  THIS ENTRY EVAL. THE CONSTRAINING EQ. ONLY            TH3
      ENTRY TH3000                                                       TH3
      ASSIGN 50 TO IGO                                                   TH3
      GO TO 5                                                            TH3
C                                                                        TH3
    4 T5 = -DRAGR - DBR*COSA                                             TH3
      T6 = LIFTR - DBR*SINA                                              TH3
    5 TCDAE = T*CODAE                                                    TH3
      TSDAE = T*SIDAE                                                    TH3
      DBCA = DB*COSA                                                     TH3
      DBSA = DB*SINA                                                     TH3
      ZZ = -TSDAE + DBSA                                                 TH3
      YY = -TCDAE + DBCA                                                 TH3
      T1 = -YY - DRAG                                                    TH3
      T2 = LIFT - ZZ                                                     TH3
      T3 = ZZ - DRAGA                                                    TH3
      T4 = LIFTA - YY                                                    TH3
      GO TO IGO                                                          TH3
C                                                                        TH3
    6 XK1VV = ASF(-DRAGVV, LIFTVV) + DRAGV**2 + LIFTV**2                 TH3
      XK1VV = XK1VV + XK1VV                                              TH3
      XK1RV = ASF(-DRAGVR, LIFTVR) - T5*DRAGV + T6*LIFTV                 TH3
      XK1RV = XK1RV + XK1RV                                              TH3
      XK1MV = T2*LIFTVM + LIFTV*LIFTM                                    TH3
      XK1MV = XK1MV + XK1MV                                              TH3
      XK1RR = ASF(-DRAGRR - DBRR*COSA, LIFTRR - DBRR*SINA) + T5**2+T6**2 TH3
      XK1RR = XK1RR + XK1RR                                              TH3
      XK1MR = T2*LIFTRM + T6*LIFTM                                       TH3
      XK1MR = XK1MR + XK1MR                                              TH3
      XK1MM = T2*LIFTMM + LIFTM**2 - (GMAX*GR)**2                        TH3
      XK1MM = XK1MM + XK1MM                                              TH3
   10 XK1VT = -CODAE*DRAGV + SIDAE*LIFTV                                 TH3
```

```
      XK1VT = XK1VT + XK1VT                                             TH3
      XK1RT = CODAE*T5 + SIDAE*T6                                       TH3
      XK1RT = XK1RT + XK1RT                                             TH3
      XK1MT = SIDAE*LIFTM                                               TH3
      XK1MT = XK1MT + XK1MT                                             TH3
      XK1VD = -TSDAE*DRAGV - TCDAE*LIFTV                                TH3
      XK1VD = XK1VD + XK1VD                                             TH3
      XK1RD = TSDAE*T5 - TCDAE*T6                                       TH3
      XK1RD = XK1RD + XK1RD                                             TH3
      XK1MD = -TCDAE*LIFTM                                              TH3
      XK1MD = XK1MD + XK1MD                                             TH3
      XK1VA = ASF(-DRAGVA, LIFTVA) - T3*DRAGV + T4*LIFTV                TH3
      XK1VA = XK1VA + XK1VA                                             TH3
      XK1RA = ASF(-DRAGRA + DBR*SINA, LIFTRA - DBR*COSA) + T3*T5 + T4*T6 TH3
      XK1RA = XK1RA + XK1RA                                             TH3
      XK1MA = T2*LIFTMA + T4*LIFTM                                      TH3
      XK1MA = XK1MA + XK1MA                                             TH3
   20 XK1V = ASF(-DRAGV, LIFTV)                                         TH3
      XK1V = XK1V + XK1V                                                TH3
      XK1R = ASF(T5, T6)                                                TH3
      XK1R = XK1R + XK1R                                                TH3
      XK1M = T2*LIFTM - GR*GMAX**2*W                                    TH3
      XK1M = XK1M + XK1M                                                TH3
   30 XK1TT = 2.                                                        TH3
      XK1TD = ASF(SIDAE, -CODAE)                                        TH3
      XK1TA = -XK1TD + LIFTA*SIDAE - DRAGA*CODAE + DB*SID               TH3
      XK1TD = XK1TD + XK1TD                                             TH3
      XK1TA = XK1TA + XK1TA                                             TH3
      XK1DD = -ASF(TCDAE, TSDAE) + T*T                                  TH3
      XK1DA = -XK1DD + T*(DB*COD - LIFTA*CODAE - DRAGA*SIDAE)           TH3
      XK1DD = XK1DD + XK1DD                                             TH3
      XK1DA = XK1DA + XK1DA                                             TH3
      XK1AA = ASF(YY - DRAGAA, ZZ + LIFTAA) + T3**2 + T4**2             TH3
      XK1AA = XK1AA + XK1AA                                             TH3
   40 XK1T - ASF(CODAE, SIDAE)                                          TH3
      XK1T = XK1T + XK1T                                                TH3
      XK1D = ASF(TSDAE, -TCDAE)                                         TH3
      XK1D = XK1D + XK1D                                                TH3
      XK1A = ASF(T3, T4)                                                TH3
      XK1A = XK1A + XK1A                                                TH3
   50 XK1 - ASF(T1, T2) - (GMAX*W)**2                                   TH3
      RETURN                                                            TH3
      END                                                               TH3
```

1277

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
COD	$\cos\delta_E$	I	See symbol		/DYNA /(	153)	DL2	I	COD
							OUTPUT	I	COD
							TH3	I	COD
							UT	M	COD
CODAE	$\cos(\alpha-\delta_E)$	I	See symbol		/DYNA /(	151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	I	See symbol		/DYNA /(	10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
DB	D_b	I	Base drag	(LBS)	/DYNA /(	163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA /(	86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA /(	87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
DRAG	D	I	Aerodynamic drag	(LBS)	/DYNA /(	69)	AL5	I	DRAG
							AL7	I	DRAG
							AL8	I	DRAG
							AL9	I	DRAG
							APPLY	I	DRAG
							CONTRL	I	DRAG
							ENVPRO	I	DRAG
							NLDRV	I	DRAG
							OUTPUT	I	DRAG
							TH3	I	DRAG
							UT	M	DRAG

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGA	$\partial D/\partial \alpha$	I	See symbol	/DYNA /	(72)	AL1	I	DRAGA
						AL5	I	DRAGA
						AL7	I	DRAGA
						AL8	I	DRAGA
						AL9	I	DRAGA
						APPLY	I	DRAGA
						TH3	I	DRAGA
						UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial \alpha^2$	I	See symbol	/DYNA /	(78)	AL1	I	DRAGAA
						AL5	I	DRAGAA
						AL7	I	DRAGAA
						AL8	I	DRAGAA
						AL9	I	DRAGAA
						APPLY	I	DRAGAA
						TH3	I	DRAGAA
						UT	M	DRAGAA
DRAGR	$\partial D/\partial R$	I	See symbol	/DYNA /	(71)	AL5	I	DRAGR
						AL7	I	DRAGR
						AL8	I	DRAGR
						AL9	I	DRAGR
						APPLY	I	DRAGR
						TH3	I	DRAGR
						UT	M	DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	I	See symbol	/DYNA /	(77)	AL1	I	DRAGRA
						AL5	I	DRAGRA
						AL7	I	DRAGRA
						AL8	I	DRAGRA
						AL9	I	DRAGRA
						APPLY	I	DRAGRA
						TH3	I	DRAGRA
						UT	M	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	I	See symbol	/DYNA /	(76)	AL5	I	DRAGRR
						AL7	I	DRAGRR
						AL8	I	DRAGRR
						AL9	I	DRAGRR
						APPLY	I	DRAGRR
						TH3	I	DRAGRR
						UT	M	DRAGRR
DRAGV	$\partial D/\partial V$	I	See symbol	/DYNA /	(70)	AL5	I	DRAGV
						AL7	I	DRAGV
						AL8	I	DRAGV
						AL9	I	DRAGV
						APPLY	I	DRAGV
						TH3	I	DRAGV
						UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	I	See symbol	/DYNA /	(75)	AL1	I	DRAGVA
						AL5	I	DRAGVA
						AL7	I	DRAGVA
						AL8	I	DRAGVA
						AL9	I	DRAGVA
						APPLY	I	DRAGVA
						TH3	I	DRAGVA
						UT	M	DRAGVA
DRAGVR	$\partial^2 D/\partial V \partial R$	I	See symbol	/DYNA /	(74)	AL5	I	DRAGVR
						AL7	I	DRAGVR
						AL8	I	DRAGVR
						AL9	I	DRAGVR
						APPLY	I	DRAGVR
						TH3	I	DRAGVR
						UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	I	See symbol	/DYNA /	(73)	AL5	I	DRAGVV
						AL7	I	DRAGVV
						AL8	I	DRAGVV
						AL9	I	DRAGVV
						APPLY	I	DRAGVV
						TH3	I	DRAGVV
						UT	M	DRAGVV

1279

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GMAX	G_{MAX}	I	Maximum total acceleration g load	/ARCDAT/(	12)	AL5	I	GMAX
						NPLANE	I	GMAX
						THROTL	I	GMAX
						TH3	I	GMAX
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	AL5	I	GR
						APPLY	I	GR
						BRANPT	I	GR
						COSTAB	I	GR
						COSTAI	I	GR
						INTRPT	I	GR
						OUTPUT	I	GR
						PDBCQL	I	GR
						QLTOSZ	I	GR
						SALVE	I	GR
						STATEF	I	GR
						TH3	I	GR
LIFT	L	I	Aerodynamic lift (LBS)	/DYNA /(	60)	AL4	I	LIFT
						AL5	I	LIFT
						AL6	I	LIFT
						APPLY	I	LIFT
						CONTRL	I	LIFT
						ENVPRO	I	LIFT
						OUTPUT	I	LIFT
						TH3	I	LIFT
						UT	O	LIFT
LIFTA	$\partial L/\partial\alpha$	I	See symbol	/DYNA /(	63)	AL1	I	LIFTA
						AL4	I	LIFTA
						AL5	I	LIFTA
						AL6	I	LIFTA
						APPLY	I	LIFTA
						TH3	I	LIFTA
						UT	O	LIFTA
LIFTAA	$\partial^2L/\partial\alpha^2$	I	See symbol	/DYNA /(	144)	AL1	I	LIFTAA
						AL4	I	LIFTAA
						AL5	I	LIFTAA
						AL6	I	LIFTAA
						APPLY	I	LIFTAA
						TH3	I	LIFTAA
						UT	O	LIFTAA
LIFTM	$\partial L/\partial m$	I	See symbol	/DYNA /(	81)	AL4	I	LIFTM
						AL5	I	LIFTM
						AL6	I	LIFTM
						APPLY	I	LIFTM
						TH3	I	LIFTM
						UT	O	LIFTM
LIFTMA	$\partial^2L/\partial m\partial\alpha$	I	See symbol	/DYNA /(	85)	AL1	I	LIFTMA
						AL4	I	LIFTMA
						AL5	I	LIFTMA
						AL6	I	LIFTMA
						APPLY	I	LIFTMA
						TH3	I	LIFTMA
						UT	O	LIFTMA
LIFTMM	$\partial^2L/\partial m^2$	I	See symbol	/DYNA /(	84)	AL4	I	LIFTMM
						AL5	I	LIFTMM
						AL6	I	LIFTMM
						APPLY	I	LIFTMM
						TH3	I	LIFTMM
						UT	O	LIFTMM
LIFTR	$\partial L/\partial R$	I	See symbol	/DYNA /(	62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	I	See symbol	/DYNA /(	68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R \partial m$	I	See symbol	/DYNA /(	83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	I	See symbol	/DYNA /(	67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	I	See symbol	/DYNA /(	61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	I	See symbol	/DYNA /(	66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTVM	$\partial^2 L/\partial V \partial m$	I	See symbol	/DYNA /(	82)	AL4	I	LIFTVM
						AL5	I	LIFTVM
						AL6	I	LIFTVM
						APPLY	I	LIFTVM
						TH3	I	LIFTVM
						UT	O	LIFTVM
LIFTVR	$\partial^2 L/\partial V \partial R$	I	See symbol	/DYNA /(	65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	I	See symbol	/DYNA /(	64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV
SID	$\sin \delta_E$	I	See symbol	/DYNA /(	154)	DL2	I	SID
						OUTPUT	I	SID
						TH3	I	SID
						UT	M	SID
SIDAE	$\sin(\alpha - \delta_E)$	I	See symbol	/DYNA /(	152)	AL1	I	SIDAE
						AL4	I	SIDAE
						AL6	I	SIDAE
						AL7	I	SIDAE
						AL8	I	SIDAE
						AL9	I	SIDAE
						APPLY	I	SIDAE
						CONTRL	I	SIDAE
						TH3	I	SIDAE
						UT	O	SIDAE

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SINA	sin α	I	See symbol		/DYNA	/(9)	AL1 AL4 AL6 AL7 AL8 AL9 APPLY CONTRL OUTPUT TH3 UT	I I I I I I I I I I M	SINA SINA SINA SINA SINA SINA SINA SINA SINA SINA SINA
T	T	I	Thrust	(LBS)	/DYNA	/(42)	ALGCON AL1 AL4 AL6 AL7 AL8 AL9 APPLY ARCIN CONTRL DL2 IMPULS OUTPUT TH1 TH2 TH3 TH4	M I I I I I I I O M I I I I I I I	T T T T T T T T T T T T T T T T T
W	W	I	Weight	(LBS)	/DYNA	/(91)	AL5 ENVPRQ OUTPUT PDBCQL QLTOSZ STATEF TH3	I I I I I M I	W W W W W W W
XK1		O	First entry of 3 word in-plane control constraints K		/MATS	/(4)	ALGCON THROTL TH1 TH2 TH3 TH4	I I O O O O	XK1 XK1 XK1 XK1 XK1 XK1
XK1MM	$K^{(I)}_{\omega\omega}$	M	The first entry in a 3x2 matrix containing $K_{y\omega}$		/MATS	/(241)	TH3	M	XK1MM
XK1MT	$K^{(I)}_{\omega T}$	M	The first entry in a 3x3 matrix containing $K_{\omega\omega}$		/MATS	/(124)	TH3	M	XK1MT
XK1RR	$K^{(I)}_{hh}$	M	The first entry in a 3x5 matrix containing K_{yh}		/MATS	/(205)	TH2 TH3 TH4	O M O	XK1RR XK1RR XK1RR
XK1RT	$K^{(I)}_{hT}$	M	The first entry in a 3x3 matrix containing $K_{h\omega}$		/MATS	/(97)	TH3	M	XK1RT
XK1T	$K^{(I)}_{T}$	M	First entry of 3x3 matrix containing the explicit partials of K with respect to ω, K_{ω}		/MATS	/(7)	ALGCON TH1 TH2 TH3	I O O M	XK1T XK1T XK1T XK1T
XK1TT	$K^{(I)}_{TT}$	O	First entry in 3x6 matrix containing $K_{\omega\omega}$		/MATS	/(23)	ALGCON TH3	I O	XKWW XK1TT
XK1V	$K^{(I)}_{V}$	M	The first entry in a 3x8 matrix containing K_{y}		/MATS	/(46)	ALGCON ALGCON TH3 TH4	I I M O	XKY XK1V XK1V XK1V
XK1VT	$K^{(I)}_{VT}$	M	The first entry in a 3x3 matrix containing $K_{v\omega}$		/MATS	/(70)	ALGCON TH3	I M	XKWY XK1VT
XK1VV	$K^{(I)}_{vv}$	M	The first entry in a 3x8 matrix containing K_{yv}		/MATS	/(142)	ALGCON TH3 TH4	I M O	XKVY XK1VV XK1VV

SUBRØUTINE
TH4

Purpose

TH4 evaluates the airbreather thrust constraint

$$T - T_{AIRB} = 0.$$

In addition, it computes the explicit first and second partials of this constraint with respect to the state and control as they are needed.

1284


```
      SUBROUTINE TH4                                                    TH4
C                                                                       TH4
C                  THIS ROUTINE APPLIES WHEN THE VEHICLE IS POWERED BY  TH4
C                  AN AIRBREATHER.                                      TH4
C                                                                       TH4
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTAPT,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ   -,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1DV ,XK2DV ,XK3DV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,   MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,   MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,   MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,   MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,   MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3M9 ,   MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,   MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,   MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,   MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,   MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1   ,PA2              MATS
      COMMON /MATS/                                                     MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)  MATS
      COMMON /MATS/                                                     MATS
```

```
     *PV      ,PG      ,PP      ,PR      ,PO      ,PVV     ,PGV     ,PPV     ,PRV     ,      MATS
     *POV     ,PGG     ,PPG     ,PRG     ,POG     ,PPP     ,PRP     ,POP     ,PRR     ,      MATS
     *POR     ,POO     ,PLG     ,PLP                                                       MATS
      EQUIVALENCE(PROD1,PROD5)                                                             MATS
      XK1VV = -TARBVV                                                                      TH4
      XK1RV = -TARBVH                                                                      TH4
      XK1RR = -TARBHH                                                                      TH4
      XK1V = -TAIRBV                                                                       TH4
      XK1R = -TAIRBH                                                                       TH4
      ENTRY TH4002                                                                         TH4
      ENTRY TH4001                                                                         TH4
      XKIT = 1.                                                                            TH4
      ENTRY TH4000                                                                         TH4
      XK1 = T - TAIRB                                                                      TH4
      RETURN                                                                               TH4
      END                                                                                  TH4
```


1285

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T	T	I	Thrust	(LBS)	/DYNA	/(42)	ALGCON	M	T
							AL1	I	T
							AL4	I	T
							AL6	I	T
							AL7	I	T
							AL8	I	T
							AL9	I	T
							APPLY	I	T
							ARCIM	O	T
							CONTRL	M	T
							DL2	I	T
							IMPULS	I	T
							OUTPUT	I	T
							TH1	I	T
							TH2	I	T
							TH3	I	T
							TH4	I	T
TAIRB		I	Air-breather engine thrust	(LBS)	/DYNA	/(219)	STATEF	I	TAIRB
							TH4	I	TAIRB
TAIRBH		I	Partial of TAIRB wrt h (altitude)		/DYNA	/(221)	TH4	I	TAIRBH
TAIRBV		I	Partial of TAIRB wrt V		/DYNA	/(220)	TH4	I	TAIRBV
TARBHH		I	Second partial of TAIRB wrt h		/DYNA	/(223)	TH4	I	TARBHH
TARBVH		I	Second partial of TAIRB wrt V and h		/DYNA	/(224)	TH4	I	TARBVH
TARBVV		I	Second partial of TAIRB wrt V		/DYNA	/(222)	TH4	I	TARBVV
XK1		O	First entry of 3 word in-plane control constraints K		/MATS	/(4)	ALGCON	I	XK1
							THROTL	I	XKI
							TH1	O	XK1
							TH2	O	XK1
							TH3	O	XK1
							TH4	O	XK1
XK1RR	$K_{hh}^{(1)}$	O	The first entry in a 3x5 matrix containing K_{yh}		/MATS	/(205)	TH2	O	XK1RR
							TH3	M	XK1RR
							TH4	O	XK1RR
XK1V	$K_{V}^{(1)}$	O	The first entry in a 3x8 matrix containing K_y		/MATS	/(46)	ALGCON	I	XKV
							ALGCON	I	XK1V
							TH3	M	XK1V
							TH4	O	XK1V
XK1VV	$K_{vv}^{(1)}$	O	The first entry in a 3x8 matrix containing K_{yv}		/MATS	/(142)	ALGCON	I	XKVV
							TH3	M	XK1VV
							TH4	O	XK1VV

SUBRØUTINE TRAJIN

1268

Purpose

TRAJIN takes care of setting various flags and paramters at the initial point of the trajectory.

TRAJIN

```
      SUBROUTINE TRAJIN                                                 TRAJIN
      COMMON /CNTRL/                                                    CNTRL
     *NU    ,ITER  ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,   CNTRL
     *KARD  ,INDX(4)      ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,   CNTRL
     *KPAGE ,NNP   ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,   CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      COMMON/GLOBAL/                                                    GLOBAL
     *GR    ,ER    ,OMGZ  ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER ,   GLOBAL
     *ITRMAX,JJOP(6)      ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20), GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      EQUIVALENCE (IPRNT, NOPRNT)                                       TRAJIN
      IPRNT = 1                                                         TRAJIN
      TSTART = TO                                                       TRAJIN
      TSTAGE = TO                                                       TRAJIN
      ALPHA = 0.                                                        TRAJIN
      DELTAE = 0.                                                       TRAJIN
      OMEGA = OMGZ                                                      TRAJIN
      OMEGA2 = OMEGA**2                                                 TRAJIN
      OMEGAT = 2.*OMEGA                                                 TRAJIN
      RETURN                                                            TRAJIN
      END                                                               TRAJIN
```

1290

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ALPHA	α	O	Angle of attack (RAD)	/DYNA /(	79)	AEROCO	I	ALPHA
						ALGCON	M	ALPHA
						AL2	I	ALPHA
						ARCIN	M	ALPHA
						CONTRL	M	ALPHA
						ENVPRO	I	ALPHA
						MOMECO	I	ALPHA
						NPLANE	I	ALPHA
						OUTPUT	I	ALPHA
						TRAJIN	O	ALPHA
						UT	I	ALPHA
						WRAPUP	I	ALPHA
DELTAE	δ_E	O	Engine deflection (RADS)	/DYNA /(	155)	ALGCON	M	DELTAE
						ARCIN	M	DELTAE
						CONTRL	M	DELTAE
						DL1	I	DELTAE
						OUTPUT	I	DELTAE
						TRAJIN	O	DELTAE
						UT	I	DELTAE
IPRNT		O	Not used.	/CNTRL /(	29)	OUTPUT	M	NOPRNT
						SALVE	O	NOPRNT
						TRAJIN	O	IPRNT
OMEGA	ω	M	Earth rotation rate (RAD/SEC)	/DYNA /(	5)	AL4	I	OMEGA
						AL7	I	OMEGA
						CONTRL	I	OMEGA
						PDBCOL	I	OMEGA
						TRAJIN	M	OMEGA
OMEGAT	2ω	O	See symbol	/DYNA /(	12)	NLDRV	I	OMEGAT
						TRAJIN	O	OMEGAT
OMEGA2	ω^2	O	See symbol	/DYNA /(	6)	AL4	I	OMEGA2
						AL7	I	OMEGA2
						AL8	I	OMEGA2
						AL9	I	OMEGA2
						NLDRV	I	OMEGA2
						TRAJIN	O	OMEGA2
OMGZ	ω	I	Earth rotation rate (RAD/SEC)	/GLOBAL/(	3)	PDBCOL	I	OMGZ
						TRAJIN	I	OMGZ
TSTAGE		O	Trajectory time at which present rocket engine ignited. (SECS)	/DYNA /(	167)	ARCIN	M	TSTAGE
						STATEF	I	TSTAGE
						TRAJIN	O	TSTAGE
TSTART		O	Trajectory time at which present subarc commenced.	/DYNA /(	141)	ARCEN	M	TSTART
						ARCIN	M	TSTART
						STATEF	I	TSTART
						TRAJIN	O	TSTART
TO	t_0	I	Trajectory start time. (SEC)	/GLOBAL/(	7)	FETCH	M	TO
						INARC	M	TO
						TRAJIN	I	TO
						WRAPUP	I	TO

SUBRØUTINE
UT

1292

Purpose

UT computes all those dynamic quantities that bear an explicit dependence on the in-plane control.

```
      SUBROUTINE UT                                                     UT
C                                                                       UT
C                 THIS ROUTINE COMPUTES ALL THOSE DYNAMIC QUANTITIES    UT
C                 WHICH BEAR AN EXPLICIT DEPENDENCE ON THE IN-PLANE     UT
C                 CONTROL                                               UT
C                                                                       UT
      REAL LCO, LSD, LCDA, LSDA                                         UT
      REAL LCDV, LCDR, LCDVV, LCDVR, LCDRR, LCDVA, LCDRA, LCDAA, LSDV,  UT
     *LSDR, LSDVV, LSDVR, LSDVA, LSDRR, LSDRA, LSDAA                    UT
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI      , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX      , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA      , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG      , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB      , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT        , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX     ,FRATE       ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R     ,G     ,SINA  ,   JUL21
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,CDE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INQF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      DIMENSION PROD1(2, 64)                                            MATS
      COMMON /MATS/                                                     MATS
     *P1    ,P2    ,P3    ,XK1   ,XK2   ,XK3   ,XK1T  ,XK2T  ,XK3T  ,   MATS
     *XK1D  ,XK2D  ,XK3D  ,XK1A  ,XK2A  ,XK3A  ,VDA   ,GDA   ,PDA   ,   MATS
     *XM19  ,XM20  ,XM21  ,XM22  ,XK1TT ,XK2TT ,XK3TT ,XK1TD ,XK2TD ,   MATS
     *XK3TD ,XK1TA ,XK2TA ,XK3TA ,XK1DD ,XK2DD ,XK3DD ,XK1DA ,XK2DA ,   MATS
     *XK3DA ,XK1AA ,XK2AA ,XK3AA ,XM41  ,XM42  ,XM43  ,XM44  ,XM45  ,   MATS
     *XK1V  ,XK2V  ,XK3V  ,XK1G  ,XK2G  ,XK3G  ,XK1P  ,XK2P  ,XK3P  ,   MATS
     *XK1R  ,XK2R  ,XK3R  ,XK1O  ,XK2O  ,XK3O  ,XK1U  ,XK2U  ,XK3U  ,   MATS
     *XK1M  ,XK2M  ,XK3M  ,XK1Z  ,XK2Z  ,XK3Z  ,XK1VT ,XK2VT ,XK3VT ,   MATS
     *XK1VD ,XK2VD ,XK3VD ,XK1VA ,XK2VA ,XK3VA ,XK1GT ,XK2GT ,XK3GT ,   MATS
     *XK1GD ,XK2GD ,XK3GD ,XK1GA ,XK2GA ,XK3GA ,XK1PT ,XK2PT ,XK3PT ,   MATS
     *XK1PD ,XK2PD ,XK3PD ,XK1PA ,XK2PA ,XK3PA ,XK1RT ,XK2RT ,XK3RT ,   MATS
     *XK1RD ,XK2RD ,XK3RD ,XK1RA ,XK2RA ,XK3RA ,XK1OT ,XK2OT ,XK3OT ,   MATS
     *XK1OD ,XK2OD ,XK3OD ,XK1OA ,XK2OA ,XK3OA ,XK1UT ,XK2UT ,XK3UT ,   MATS
     *XK1UD ,XK2UD ,XK3UD ,XK1UA ,XK2UA ,XK3UA ,XK1MT ,XK2MT ,XK3MT     MATS
      COMMON /MATS/                                                     MATS
     *XK1MD ,XK2MD ,XK3MD ,XK1MA ,XK2MA ,XK3MA ,XK1ZT ,XK2ZT ,XK3ZT ,   MATS
     *XK1ZD ,XK2ZD ,XK3ZD ,XK1ZA ,XK2ZA ,XK3ZA ,XK1VV ,XK2VV ,XK3VV ,   MATS
     *XK1GV ,XK2GV ,XK3GV ,XK1PV ,XK2PV ,XK3PV ,XK1RV ,XK2RV ,XK3RV ,   MATS
     *XK1OV ,XK2OV ,XK3OV ,XK1UV ,XK2UV ,XK3UV ,XK1MV ,XK2MV ,XK3MV ,   MATS
```

1294

```
     *XK1ZV ,XK2ZV ,XK3ZV ,XK1GG ,XK2GG ,XK3GG ,XK1PG ,XK2PG ,XK3PG ,      MATS
     *XK1RG ,XK2RG ,XK3RG ,XK1OG ,XK2OG ,XK3OG ,XK1UG ,XK2UG ,XK3UG ,      MATS
     *XK1MG ,XK2MG ,XK3MG ,XK1ZG ,XK2ZG ,XK3ZG ,XK1PP ,XK2PP ,XK3PP ,      MATS
     *XK1RP ,XK2RP ,XK3RP ,XK1OP ,XK2OP ,XK3OP ,XK1UP ,XK2UP ,XK3UP ,      MATS
     *XK1MP ,XK2MP ,XK3MP ,XK1ZP ,XK2ZP ,XK3ZP ,XK1RR ,XK2RR ,XK3RR ,      MATS
     *XK1OR ,XK2OR ,XK3OR ,XK1UR ,XK2UR ,XK3UR ,XK1MR ,XK2MR ,XK3MR ,      MATS
     *XK1ZR ,XK2ZR ,XK3ZR ,XK1OO ,XK2OO ,XK3OO ,XK1UO ,XK2UO ,XK3UO ,      MATS
     *XK1MO ,XK2MO ,XK3MO ,XK1ZO ,XK2ZO ,XK3ZO ,XK1UU ,XK2UU ,XK3UU ,      MATS
     *XK1MU ,XK2MU ,XK3MU ,XK1ZU ,XK2ZU ,XK3ZU ,XK1MM ,XK2MM ,XK3MM ,      MATS
     *XK1ZM ,XK2ZM ,XK3ZM ,XK1ZZ ,XK2ZZ ,XK3ZZ ,XKPI11,XKPI21,XKPI31,      MATS
     *XKPI12,XKPI22,XKPI32,XKPI13,XKPI23,XKPI33,PA1    ,PA2               MATS
      COMMON /MATS/                                                       MATS
     *DPDY(3, 8), DEPDEY(2, 8), DPDL(3, 3), PROD5(3, 64), PROD9(2, 24)    MATS
      COMMON /MATS/                                                       MATS
     *PV    ,PG    ,PP    ,PR    ,PO    ,PVV   ,PGV   ,PPV   ,PRV   ,      MATS
     *POV   ,PGG   ,PPG   ,PRG   ,POG   ,PPP   ,PRP   ,POP   ,PRR   ,      MATS
     *POR   ,POO   ,PLG   ,PLP                                            MATS
      EQUIVALENCE(PROD1,PROD5)                                            MATS
C                     THIS ENTRY COMPUTES ALL THE AFORESAID QUANTITIES    UT
C                     INCLUDING  ALL THEIR 2ND, 1ST AND MIXED PARTIALS    UT
C                     W/RESP. TO STATE AND CONTROL.  FOR THESE REASONS,   UT
C                     IT IS ONLY CALLED AFTER THE IN-PLANE CONTROL HAS    UT
C                     BEEN CONVERGED.                                     UT
      ASSIGN 1041 TO LABL3                                                UT
      ASSIGN 1071 TO LABL4                                                UT
      ASSIGN 1061 TO LABL5                                                UT
      GO TO 100                                                           UT   100
C                     THIS ENTRY COMPUTES ALL THOSE CONTROL-DEPENDENT     UT
C                     QUANTITIES THAT ARE NEEDED TO CONVERGE THE IN-PLANE UT
C                     CONTROL WHEN THE ANGLE OF ATTACK IS TO BE OPTIMAL   UT
      ENTRY UTOP                                                          UT
      ASSIGN 105 TO LABL3                                                 UT
      ASSIGN 108 TO LABL4                                                 UT
      ASSIGN 1062 TO LABL5                                                UT
      GO TO 100                                                           UT   100
C                     THIS ENTRY COMPUTES ALL THOSE CONTROL-DEPENDENT     UT
C                     QUANTITIES THAT ARE NEEDED TO CONVERGE THE IN-PLANE UT
C                     CONTROL WHEN THE ANGLE OF ATTACK IS TO BE NON-      UT
C                     OPTIMAL                                             UT
      ENTRY UTNOP                                                         UT
      ASSIGN 106 TO LABL3                                                 UT
      ASSIGN 109 TO LABL4                                                 UT
      ASSIGN 1063 TO LABL5                                                UT
  100 SINA = SIN(ALPHA)                                                   UT
      COSA = COS(ALPHA)                                                   UT
      SID = SIN(DELTAE)                                                   UT
      COD = COS(DELTAE)                                                   UT
      SIDAE = SINA*COD - COSA*SID                                         UT
      CODAE = COSA*COD + SINA*SID                                         UT
  104 IF(.NOT.ILOAD) RETURN                                               UT
      CALL AEROCO                                                         UT
      QS = Q*SREF                                                         UT
      GO TO LABL3                                                         UT
 1041 QMV = Q*MACHV                                                       UT
      QMR = Q*MACHR                                                       UT
      QMVMV = QMV*MACHV                                                   UT
      QMVMR = QMV*MACHR                                                   UT
      QMRMR = QMR*MACHR                                                   UT
      QVMV = QV*MACHV*2.                                                  UT
      QVMR = QV*MACHR + QR*MACHV + Q*MACHVR                               UT
      QRMR = QR*MACHR*2. + Q*MACHRR                                       UT
      ULFTVV = SREF*(QVV*CL + QVMV*CLM + QMVMV*CLMM)                      UT
      DRAGVV = SREF*(QVV*CD + QVMV*CDM + QMVMV*CDMM)                      UT
      ULFTVR = SREF*(QVR*CL + QVMR*CLM + QMVMR*CLMM)                      UT
      DRAGVR = SREF*(QVR*CD + QVMR*CDM + QMVMR*CDMM)                      UT
      ULFTRR = SREF*(QRR*CL + QRMR*CLM + QMRMR*CLMM)                      UT
      DRAGRR = SREF*(QRR*CD + QRMR*CDM + QMRMR*CDMM)                      UT
      ULFTVA = SREF*(QV*CLA + QMV*CLAM)                                   UT
      DRAGVA = SREF*(QV*CDA + QMV*CDAM)                                   UT
      ULFTRA = SREF*(QR*CLA + QMR*CLAM)                                   UT
      DRAGRA = SREF*(QR*CDA + QMR*CDAM)                                   UT
      ULFTV  = SREF*(QV*CL + QMV*CLM)                                     UT
      DRAGV  = SREF*(QV*CD + QMV*CDM)                                     UT
      ULFTR  = SREF*(QR*CL + QMR*CLM)                                     UT
      DRAGR  = SREF*(QR*CD + QMR*CDM)                                     UT
```

```
      105 ULFTAA = QS*CLAA                                              UT
          DRAGAA = QS*CDAA                                              UT
      106 ULFTA = QS*CLA                                                UT
          DRAGA = QS*CDA                                                UT
          ULFT = QS*CL                                                  UT
          DRAG = QS*CD                                                  UT
          IF(JAER .EQ. 3) GO TO 107                                     UT   107
          GO TO LABL5                                                   UT
     1061 LIFTVV = ULFTVV                                               UT
          LIFTVR = ULFTVR                                               UT
          LIFTRR = ULFTRR                                               UT
          LIFTVA = ULFTVA                                               UT
          LIFTRA = ULFTRA                                               UT
          LIFTV  = ULFTV                                                UT
          LIFTR  = ULFTR                                                UT
     1062 LIFTAA = ULFTAA                                               UT
     1063 LIFTA  = ULFTA                                                UT
          LIFT = ULFT                                                   UT
          RETURN                                                        UT
      107 CALL MOMECO                                                   UT
          QSD = QS*DREF                                                 UT
          LCD = ULFT*COSA + DRAG*SINA                                   UT
          LSD = ULFT*SINA - DRAG*COSA                                   UT
          LCDA = ULFTA*COSA + DRAGA*SINA                                UT
          LSDA = ULFTA*SINA - DRAGA*COSA                                UT
          XF = XCG - XCGR                                               UT
          ZF = ZCG - ZCGR                                               UT
          YF = ZE - ZCG                                                 UT
          WF = XT - XCG                                                 UT
          FACTOR = XJ/WF                                                UT
          LCDAA = LCDA - LSD                                            UT
          LSDAA = LSDA + LCD                                            UT
          XMCGA = LCDAA*XF + LSDAA*ZF + QSD*CMA                         UT
          XMCG = LCD*XF + LSD*ZF + DB*YF + QSD*CM                       UT
          GO TO LABL4                                                   UT
          SD = SREF*DREF                                                UT
     1071 LCDV = COSA*ULFTV + SINA*DRAGV                                UT
          LCDR = COSA*ULFTR + SINA*DRAGR                                UT
          LSDV = SINA*ULFTV - COSA*DRAGV                                UT
          LSDR = SINA*ULFTR - COSA*DRAGR                                UT
          LCDVV = COSA*ULFTVV + SINA*DRAGVV                             UT
          LCDVR = COSA*ULFTVR + SINA*DRAGVR                             UT
          LCDVA = COSA*ULFTVA + SINA*DRAGVA - LSDV                      UT
          LCDRR = COSA*ULFTRR + SINA*DRAGRR                             UT
          LCDRA = COSA*ULFTRA + SINA*DRAGRA - LSDR                      UT
          LSDVV = SINA*ULFTVV - COSA*DRAGVV                             UT
          LSDVR = SINA*ULFTVR - COSA*DRAGVR                             UT
          LSDVA = SINA*ULFTVA - COSA*DRAGVA + LCDV                      UT
          LSDRR = SINA*ULFTRR - COSA*DRAGRR                             UT
          LSDRA = SINA*ULFTRA - COSA*DRAGRA + LCDR                      UT
          XMCGVV = SD*(QVV*CM + QVMV*CMM + QMVMV*CMMM) + LCDVV*XF + LSDVV*ZF UT
          XMCGVR = SD*(QVR*CM + QVMR*CMM + QMVMR*CMMM) + LCDVR*XF + LSDVR*ZF UT
          XMCGRR = SD*(QRR*CM + QRMR*CMM + QMRMR*CMMM) + LCDRR*XF + LSDRR*ZF UT
         * + DBRR*YF                                                    UT
          XMCGVM = LCDV*XCGM + LSDV*ZCGM                                UT
          XMCGRM = LCDR*XCGM + (LSDR - DBR)*ZCGM                        UT
          XMCGMM = LCD*XCGMM + (LSD - DB)*ZCGMM                         UT
          XMCGVA = SD*(QV*CMA + QMV*CMAM) + LCDVA*XF + LSDVA*ZF         UT.
          XMCGRA = SD*(QR*CMA + QMR*CMAM) + LCDRA*XF + LSDRA*ZF         UT
          XMCGMA = LCDAA*XCGM + LSDAA*ZCGM                              UT
          XMCGV = SD*(QV*CM + QMV*CMM) + LCDV*XF + LSDV*ZF              UT
          XMCGR = SD*(QR*CM + QMR*CMM) + LCDR*XF + LSDR*ZF + DBR*YF     UT
          XMCGM = LCD*XCGM + (LSD - DB)*ZCGM                            UT
          XCGMWF = XCGM/WF                                              UT
          FACTV = XJV/WF                                                UT
          FACTR = XJR/WF                                                UT
          FACTM = FACTOR*XCGMWF                                         UT
          FACTVV = XJVV/WF                                              UT
          FACTVR = XJVR/WF                                              UT
          FACTVM = FACTV*XCGMWF                                         UT
          FACTRR = XJRR/WF                                              UT
          FACTRM = FACTR*XCGMWF                                         UT
          FACTMM = FACTOR/WF*(XCGMM + 2.*XCGM*XCGMWF)                   UT
          LIFTVV = ULFTVV + FACTVV*XMCG + 2.*FACTV*XMCGV + FACTOR*XMCGVV    UT
          LIFTRR = ULFTRR + FACTRR*XMCG + 2.*FACTR*XMCGR + FACTOR*XMCGRR    UT
```

```
      LIFTVR = ULFTVR + FACTVR*XMCG + FACTV*XMCGR + FACTR*XMCGV           UT
     * + FACTOR*XMCGVR                                                    UT
      LIFTVM = FACTVM*XMCG + FACTOR*XMCGVM + FACTV*XMCGM + FACTM*XMCGV    UT
      LIFTRM = FACTRM*XMCG + FACTOR*XMCGRM + FACTR*XMCGM + FACTM*XMCGR    UT
      LIFTMM = FACTMM*XMCG + FACTOR*XMCGMM + FACTM*XMCGM*2.               UT
      LIFTVA = FACTOR*XMCGVA + FACTV*XMCGA + ULFTVA                       UT
      LIFTRA = FACTOR*XMCGRA + FACTR*XMCGA + ULFTRA                       UT
      LIFTMA = FACTOR*XMCGMA + FACTM*XMCGA                                UT
      LIFTV = FACTOR*XMCGV + FACTV*XMCG + ULFTV                           UT
      LIFTR = FACTOR*XMCGR + FACTR*XMCG + ULFTR                           UT
      LIFTM = FACTOR*XMCGM + FACTM*XMCG                                   UT
  108 XMCGAA = (ULFTAA*COSA + DRAGAA*SINA - LSDA - LSDA - LCD)*XF         UT
     * + (ULFTAA*SINA - DRAGAA*COSA + LCDA + LCDA - LSD )*ZF + QSD*CMAA   UT
      LIFTAA = ULFTAA + FACTOR*XMCGAA                                     UT
  109 LIFTA = ULFTA + FACTOR*XMCGA                                        UT
      LIFT = ULFT + FACTOR*XMCG                                           UT
      RETURN                                                              UT
      END                                                                 UT
```

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	I	Angle of attack	(RAD)	/DYNA	/(79)	AEROCO ALGCON AL2 ARCIN CONTRL ENVPRO MOMECO NPLANE OUTPUT TRAJIN UT WRAPUP	I M I M M I I I I O I I	ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA
CD	C_D	I	Drag coefficient		/DYNA	/(192)	AEROCO OUTPUT UT	O I I	CD CD CD
CDA	C_{D_α}	I	Drag coefficient slope	(RAD^{-1})	/DYNA	/(193)	AEROCO UT	M I	CDA CDA
CDAA	$\partial C_{D_\alpha}/\partial\alpha$	I	See symbol		/DYNA	/(195)	AEROCO UT	M I	CDAA CDAA
CDAM	$\partial C_{D_\alpha}/\partial M$	I	See symbol		/DYNA	/(197)	AEROCO UT	M I	CDAM CDAM
CDM	$\partial C_D/\partial M$	I	See symbol		/DYNA	/(194)	AEROCO UT	O I	CDM CDM
CDMM	$\partial^2 C_D/\partial M^2$	I	See symbol		/DYNA	/(196)	AEROCO UT	O I	CDMM CDMM
CL	C_L	I	Lift coefficient		/DYNA	/(186)	AEROCO OUTPUT UT	M I I	CL CL CL
CLA	C_{L_α}	I	Lift coefficient slope	(RAD^{-1})	/DYNA	/(187)	AEROCO STATEF UT	M M I	CLA CLA CLA
CLAA	$\partial C_{L_\alpha}/\partial\alpha$	I	See symbol		/DYNA	/(189)	AEROCO UT	M I	CLAA CLAA
CLAM	$\partial C_{L_\alpha}/\partial M$	I	See symbol		/DYNA	/(191)	AEROCO STATEF UT	M M I	CLAM CLAM CLAM
CLM	$\partial C_L/\partial M$	I	See symbol		/DYNA	/(188)	AEROCO UT	M I	CLM CLM
CLMM	$\partial^2 C_L/\partial M^2$	I	See symbol		/DYNA	/(190)	AEROCO UT	M I	CLMM CLMM
CM	C_m	I	Moment coefficient		/DYNA	/(122)	MOMECO UT	O I	CM CM
CMA	C_{m_α}	I	Moment coefficient slope	(RAD^{-1})	/DYNA	/(123)	MOMECO STATEF UT	I M I	CMA CMA CMA
CMAA	$\partial C_{m_\alpha}/\partial\alpha$	I	See symbol		/DYNA	/(125)	MOMECO UT	O I	CMAA CMAA
CMAM	$\partial C_{m_\alpha}/\partial M$	I	See symbol		/DYNA	/(127)	MOMECO STATEF UT	I M I	CMAM CMAM CMAM
CMM	$\partial C_m/\partial M$	I	See symbol		/DYNA	/(124)	MOMECO UT	O I	CMM CMM
CMMM	$\partial^2 C_m/\partial M^2$	I	See symbol		/DYNA	/(126)	MOMECO UT	O I	CMMM CMMM
COD	$\cos\delta_E$	M	See symbol		/DYNA	/(153)	DL2 OUTPUT TH3 UT	I I I M	COD COD COD COD

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CODAE	$\cos(\alpha-\delta_E)$	O	See symbol		/DYNA	/(151)	AL1	I	CODAE
							AL4	I	CODAE
							AL6	I	CODAE
							AL7	I	CODAE
							AL8	I	CODAE
							AL9	I	CODAE
							APPLY	I	CODAE
							CONTRL	I	CODAE
							NLDRV	I	CODAE
							TH3	I	CODAE
							UT	O	CODAE
COSA	$\cos\alpha$	M	See symbol		/DYNA	/(10)	AL1	I	COSA
							AL4	I	COSA
							AL6	I	COSA
							AL7	I	COSA
							AL8	I	COSA
							AL9	I	COSA
							APPLY	I	COSA
							CONTRL	I	COSA
							NLDRV	I	COSA
							OUTPUT	I	COSA
							TH3	I	COSA
							UT	M	COSA
DB	D_b	I	Base drag	(LBS)	/DYNA	/(163)	AL1	I	DB
							AL4	I	DB
							AL6	I	DB
							AL7	I	DB
							AL8	I	DB
							AL9	I	DB
							APPLY	I	DB
							CONTRL	I	DB
							NLDRV	I	DB
							OUTPUT	I	DB
							STATEF	I	DB
							TH3	I	DB
							UT	I	DB
DBR	$\partial D_b/\partial R$	I	See symbol		/DYNA	/(86)	AL1	I	DBR
							AL4	I	DBR
							AL6	I	DBR
							AL7	I	DBR
							AL8	I	DBR
							AL9	I	DBR
							APPLY	I	DBR
							STATEF	I	DBR
							TH3	I	DBR
							UT	I	DBR
DBRR	$\partial^2 D_b/\partial R^2$	I	See symbol		/DYNA	/(87)	AL4	I	DBRR
							AL6	I	DBRR
							AL7	I	DBRR
							AL8	I	DBRR
							AL9	I	DBRR
							APPLY	I	DBRR
							STATEF	I	DBRR
							TH3	I	DBRR
							UT	I	DBRR
DELTAE	δ_E	I	Engine deflection	(RADS)	/DYNA	/(155)	ALGCON	M	DELTAE
							ARCIN	M	DELTAE
							CONTRL	M	DELTAE
							DL1	I	DELTAE
							OUTPUT	I	DELTAE
							TRAJIN	O	DELTAE
							UT	I	DELTAE

1299

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
DRAG	D	M	Aerodynamic drag	(LBS)	/DYNA	/(69)	AL5	I	DRAG
							AL7	I	DRAG
							AL8	I	DRAG
							AL9	I	DRAG
							APPLY	I	DRAG
							CONTRL	I	DRAG
							ENVPRO	I	DRAG
							NLDRV	I	DRAG
							OUTPUT	I	DRAG
							TH3	I	DRAG
							UT	M	DRAG
DRAGA	$\partial D/\partial \alpha$	M	See symbol		/DYNA	/(72)	AL1	I	DRAGA
							AL5	I	DRAGA
							AL7	I	DRAGA
							AL8	I	DRAGA
							AL9	I	DRAGA
							APPLY	I	DRAGA
							TH3	I	DRAGA
							UT	M	DRAGA
DRAGAA	$\partial^2 D/\partial \alpha^2$	M	See symbol		/DYNA	/(78)	AL1	I	DRAGAA
							AL5	I	DRAGAA
							AL7	I	DRAGAA
							AL8	I	DRAGAA
							AL9	I	DRAGAA
							APPLY	I	DRAGAA
							TH3	I	DRAGAA
							UT	M	DRAGAA
DRAGR	$\partial D/\partial R$	M	See symbol		/DYNA	/(71)	AL5	I	DRAGR
							AL7	I	DRAGR
							AL8	I	DRAGR
							AL9	I	DRAGR
							APPLY	I	DRAGR
							TH3	I	DRAGR
							UT	M	DRAGR
DRAGRA	$\partial^2 D/\partial R \partial \alpha$	M	See symbol		/DYNA	/(77)	AL1	I	DRAGRA
							AL5	I	DRAGRA
							AL7	I	DRAGRA
							AL8	I	DRAGRA
							AL9	I	DRAGRA
							APPLY	I	DRAGRA
							TH3	I	DRAGRA
							UT	M	DRAGRA
DRAGRR	$\partial^2 D/\partial R^2$	M	See symbol		/DYNA	/(76)	AL5	I	DRAGRR
							AL7	I	DRAGRR
							AL8	I	DRAGRR
							AL9	I	DRAGRR
							APPLY	I	DRAGRR
							TH3	I	DRAGRR
							UT	M	DRAGRR
DRAGV	$\partial D/\partial V$	M	See symbol		/DYNA	/(70)	AL5	I	DRAGV
							AL7	I	DRAGV
							AL8	I	DRAGV
							AL9	I	DRAGV
							APPLY	I	DRAGV
							TH3	I	DRAGV
							UT	M	DRAGV
DRAGVA	$\partial^2 D/\partial V \partial \alpha$	M	See symbol		/DYNA	/(75)	AL1	I	DRAGVA
							AL5	I	DRAGVA
							AL7	I	DRAGVA
							AL8	I	DRAGVA
							AL9	I	DRAGVA
							APPLY	I	DRAGVA
							TH3	I	DRAGVA
							UT	M	DRAGVA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
DRAGVR	$\partial^2 D/\partial V \partial R$	M	See symbol		/DYNA /(	74)	AL5	I	DRAGVR
							AL7	I	DRAGVR
							AL8	I	DRAGVR
							AL9	I	DRAGVR
							APPLY	I	DRAGVR
							TH3	I	DRAGVR
							UT	M	DRAGVR
DRAGVV	$\partial^2 D/\partial V^2$	M	See symbol		/DYNA /(	73)	AL5	I	DRAGVV
							AL7	I	DRAGVV
							AL8	I	DRAGVV
							AL9	I	DRAGVV
							APPLY	I	DRAGVV
							TH3	I	DRAGVV
							UT	M	DRAGVV
DREF	$\bar{D}_{ref}$	I	Aerodynamic reference length		/ARCDAT/(	37)	STATEF	I	DREF
							UT	I	DREF
ILOAD		I	Logical flag that is true if there is any aerodynamic load on the vehicle.		/DYNA /(	181)	ARCIN	M	ILOAD
							CONTRL	I	ILOAD
							NPLANE	I	ILOAD
							UT	I	ILOAD
JAER		I	Aerodynamic model option flag		/ARCDAT/(	9)	AEROCO	I	JAER
							ARCIN	I	JAER
							OUTPUT	I	JAER
							STATEF	I	JAER
							UT	I	JAER
LIFT	L	O	Aerodynamic lift	(LBS)	/DYNA /(	60)	AL4	I	LIFT
							AL5	I	LIFT
							AL6	I	LIFT
							APPLY	I	LIFT
							CONTRL	I	LIFT
							ENVPRD	I	LIFT
							OUTPUT	I	LIFT
							TH3	I	LIFT
							UT	O	LIFT
LIFTA	$\partial L/\partial \alpha$	O	See symbol		/DYNA /(	63)	AL1	I	LIFTA
							AL4	I	LIFTA
							AL5	I	LIFTA
							AL6	I	LIFTA
							APPLY	I	LIFTA
							TH3	I	LIFTA
							UT	O	LIFTA
LIFTAA	$\partial^2 L/\partial \alpha^2$	O	See symbol		/DYNA /(	144)	AL1	I	LIFTAA
							AL4	I	LIFTAA
							AL5	I	LIFTAA
							AL6	I	LIFTAA
							APPLY	I	LIFTAA
							TH3	I	LIFTAA
							UT	O	LIFTAA
LIFTM	$\partial L/\partial m$	O	See symbol		/DYNA /(	81)	AL4	I	LIFTM
							AL5	I	LIFTM
							AL6	I	LIFTM
							APPLY	I	LIFTM
							TH3	I	LIFTM
							UT	O	LIFTM
LIFTMA	$\partial^2 L/\partial m \partial \alpha$	O	See symbol		/DYNA /(	85)	AL1	I	LIFTMA
							AL4	I	LIFTMA
							AL5	I	LIFTMA
							AL6	I	LIFTMA
							APPLY	I	LIFTMA
							TH3	I	LIFTMA
							UT	O	LIFTMA
LIFTMM	$\partial^2 L/\partial m^2$	O	See symbol		/DYNA /(	84)	AL4	I	LIFTMM
							AL5	I	LIFTMM
							AL6	I	LIFTMM
							APPLY	I	LIFTMM
							TH3	I	LIFTMM
							UT	O	LIFTMM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
LIFTR	$\partial L/\partial R$	O	See symbol	/DYNA /(	62)	AL4	I	LIFTR
						AL5	I	LIFTR
						AL6	I	LIFTR
						APPLY	I	LIFTR
						TH3	I	LIFTR
						UT	O	LIFTR
LIFTRA	$\partial^2 L/\partial R \partial \alpha$	O	See symbol	/DYNA /(	68)	AL1	I	LIFTRA
						AL4	I	LIFTRA
						AL5	I	LIFTRA
						AL6	I	LIFTRA
						APPLY	I	LIFTRA
						TH3	I	LIFTRA
						UT	O	LIFTRA
LIFTRM	$\partial^2 L/\partial R \partial m$	O	See symbol	/DYNA /(	83)	AL4	I	LIFTRM
						AL5	I	LIFTRM
						AL6	I	LIFTRM
						APPLY	I	LIFTRM
						TH3	I	LIFTRM
						UT	O	LIFTRM
LIFTRR	$\partial^2 L/\partial R^2$	O	See symbol	/DYNA /(	67)	AL4	I	LIFTRR
						AL5	I	LIFTRR
						AL6	I	LIFTRR
						APPLY	I	LIFTRR
						TH3	I	LIFTRR
						UT	O	LIFTRR
LIFTV	$\partial L/\partial V$	O	See symbol	/DYNA /(	61)	AL4	I	LIFTV
						AL5	I	LIFTV
						AL6	I	LIFTV
						APPLY	I	LIFTV
						TH3	I	LIFTV
						UT	O	LIFTV
LIFTVA	$\partial^2 L/\partial V \partial \alpha$	O	See symbol	/DYNA /(	66)	AL1	I	LIFTVA
						AL4	I	LIFTVA
						AL5	I	LIFTVA
						AL6	I	LIFTVA
						APPLY	I	LIFTVA
						TH3	I	LIFTVA
						UT	O	LIFTVA
LIFTVM	$\partial^2 L/\partial V \partial m$	O	See symbol	/DYNA /(	82)	AL4	I	LIFTVM
						AL5	I	LIFTVM
						AL6	I	LIFTVM
						APPLY	I	LIFTVM
						TH3	I	LIFTVM
						UT	O	LIFTVM
LIFTVR	$\partial^2 L/\partial V \partial R$	O	See symbol	/DYNA /(	65)	AL4	I	LIFTVR
						AL5	I	LIFTVR
						AL6	I	LIFTVR
						APPLY	I	LIFTVR
						TH3	I	LIFTVR
						UT	O	LIFTVR
LIFTVV	$\partial^2 L/\partial V^2$	O	See symbol	/DYNA /(	64)	AL4	I	LIFTVV
						AL5	I	LIFTVV
						AL6	I	LIFTVV
						APPLY	I	LIFTVV
						TH3	I	LIFTVV
						UT	O	LIFTVV
MACHR	$\partial M/\partial R$	I	See symbol	/DYNA /(	44)	STATEF	M	MACHR
						UT	I	MACHR
MACHRR	$\partial^2 M/\partial R^2$	I	See symbol	/DYNA /(	118)	STATEF	M	MACHRR
						UT	I	MACHRR
MACHV	$\partial M/\partial V$	I	See symbol	/DYNA /(	43)	STATEF	M	MACHV
						UT	I	MACHV
MACHVR	$\partial^2 M/\partial V \partial R$	I	See symbol	/DYNA /(	117)	STATEF	O	MACHVR
						UT	I	MACHVR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
Q	q	I	Dynamic pressure	(LBS/FT^2)	/DYNA /(	27)	ENVPRO	I	Q
							OUTPUT	I	Q
							PDBCOL	I	Q
							STATEF	M	Q
							UT	I	Q
QR	$\partial q/\partial R$	I	See symbol		/DYNA /(	29)	PDBCOL	I	QR
							STATEF	M	QR
							UT	I	QR
QRR	$\partial^2 q/\partial R^2$	I	See symbol		/DYNA /(	32)	STATEF	M	QRR
							UT	I	QRR
QV	$\partial q/\partial V$	I	See symbol		/DYNA /(	28)	PDBCOL	I	QV
							STATEF	M	QV
							UT	I	QV
QVR	$\partial^2 q/\partial V \partial R$	I	See symbol		/DYNA /(	31)	STATEF	M	QVR
							UT	I	QVR
QVV	$\partial^2 q/\partial V^2$	I	See symbol		/DYNA /(	30)	STATEF	M	QVV
							UT	I	QVV
SID	$\sin \delta_E$	M	See symbol		/DYNA /(	154)	DL2	I	SID
							OUTPUT	I	SID
							TH3	I	SID
							UT	M	SID
SIDAE	$\sin(\alpha - \delta_E)$	O	See symbol		/DYNA /(	152)	AL1	I	SIDAE
							AL4	I	SIDAE
							AL6	I	SIDAE
							AL7	I	SIDAE
							AL8	I	SIDAE
							AL9	I	SIDAE
							APPLY	I	SIDAE
							CONTRL	I	SIDAE
							TH3	I	SIDAE
							UT	O	SIDAE
SINA	$\sin \alpha$	M	See symbol		/DYNA /(	9)	AL1	I	SINA
							AL4	I	SINA
							AL6	I	SINA
							AL7	I	SINA
							AL8	I	SINA
							AL9	I	SINA
							APPLY	I	SINA
							CONTRL	I	SINA
							OUTPUT	I	SINA
							TH3	I	SINA
							UT	M	SINA
SREF	S_{ref}	I	Aerodynamic reference area	(FT^2)	/ARCDAT/(	1)	ARCIN	I	SREF
							BNDRY	I	ARCDA
							CHECK	I	ARCDA
							FETCH	I	ARCDA
							SALVE	I	ARCDA
							STATEF	I	SREF
							UT	I	SREF
							WRAPUP	I	ARCDA
ULFT	L_u	M	Untrimmed aerodynamic lift	(LBS)	/DYNA /(	164)	AL3	I	ULFT
							NPLANE	I	ULFT
							UT	M	ULFT
ULFTA	$\partial L_u/\partial \alpha$	M	See symbol		/DYNA /(	166)	AL3	I	ULFTA
							UT	M	ULFTA
ULFTAA	$\partial^2 L_u/\partial \alpha^2$	M	See symbol		/DYNA /(	178)	AL3	I	ULFTAA
							UT	M	ULFTAA
ULFTR	$\partial L_u/\partial R$	M	See symbol		/DYNA /(	133)	AL3	I	ULFTR
							UT	M	ULFTR
ULFTRA	$\partial^2 L_u/\partial R \partial \alpha$	M	See symbol		/DYNA /(	138)	AL3	I	ULFTRA
							UT	M	ULFTRA
ULFTRR	$\partial^2 L_u/\partial R^2$	M	See symbol		/DYNA /(	137)	AL3	I	ULFTRR
							UT	M	ULFTRR
ULFTV	$\partial L_u/\partial V$	M	See symbol		/DYNA /(	132)	AL3	I	ULFTV
							UT	M	ULFTV

303

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ULFTVA	$\partial^2 L_u/\partial V \partial \alpha$	M	See symbol	/DYNA /(	136)	AL3 UT	I M	ULFTVA ULFTVA
ULFTVR	$\partial^2 L_u/\partial V \partial R$	M	See symbol	/DYNA /(	135)	AL3 UT	I M	ULFTVR ULFTVR
ULFTVV	$\partial^2 L_u/\partial V^2$	M	See symbol	/DYNA /(	134)	AL3 UT	I M	ULFTVV ULFTVV
XCG	X_{CG}	I	Center of gravity body x station (FT)	/DYNA /(	157)	DL2 STATEF UT	I I I	XCG XCG XCG
XCGM	$\partial X_{CG}/\partial m$	I	See symbol	/DYNA /(	108)	DL2 STATEF UT	I M I	XCGM XCGM XCGM
XCGMM	$\partial^2 X_{CG}/\partial m^2$	I	See symbol	/DYNA /(	109)	DL2 STATEF UT	I M I	XCGMM XCGMM XCGMM
XCGR	X_{CGR}	I	Reference xcg location (FT)	/ARCDAT/(	32)	UT	I	XCGR
XJ	$\dot{j}$	I	Control blend factor	/DYNA /(	159)	ARCIN DL2 OUTPUT STATEF UT	O I I I I	XJ XJ XJ XJ XJ
XJR	$\partial \dot{j}/\partial R$	I	See symbol	/DYNA /(	113)	DL2 STATEF UT	I O I	XJR XJR XJR
XJRR	$\partial^2 \dot{j}/\partial R^2$	I	See symbol	/DYNA /(	116)	DL2 STATEF UT	I O I	XJRR XJRR XJRR
XJV	$\partial \dot{j}/\partial V$	I	See symbol	/DYNA /(	112)	DL2 STATEF UT	I O I	XJV XJV XJV
XJVR	$\partial^2 \dot{j}/\partial V \partial R$	I	See symbol	/DYNA /(	115)	DL2 STATEF UT	I O I	XJVR XJVR XJVR
XJVV	$\partial^2 \dot{j}/\partial V^2$	I	See symbol	/DYNA /(	114)	DL2 STATEF UT	I O I	XJVV XJVV XJVV
XMCG	M_{CG}	M	Aerodynamic moment about center of gravity (FT-LBS)	/DYNA /(	160)	DL2 OUTPUT UT	I I M	XMCG XMCG XMCG
XMCGA	$\partial M_{CG}/\partial \alpha$	M	See symbol	/DYNA /(	176)	DL2 UT	I M	XMCGA XMCGA
XMCGAA	$\partial^2 M_{CG}/\partial \alpha^2$	M	See symbol	/DYNA /(	169)	DL2 UT	I M	XMCGAA XMCGAA
XMCGM	$\partial M_{CG}/\partial m$	M	See symbol	/DYNA /(	203)	DL2 UT	I M	XMCGM XMCGM
XMCGMA	$\partial^2 M_{CG}/\partial m \partial \alpha$	M	See symbol	/DYNA /(	212)	DL2 UT	I M	XMCGMA XMCGMA
XMCGMM	$\partial^2 M_{CG}/\partial m^2$	M	See symbol	/DYNA /(	211)	DL2 UT	I M	XMCGMM XMCGMM
XMCGR	$\partial M_{CG}/\partial R$	M	See symbol	/DYNA /(	202)	DL2 UT	I M	XMCGR XMCGR
XMCGRA	$\partial^2 M_{CG}/\partial R \partial \alpha$	M	See symbol	/DYNA /(	210)	DL2 UT	I M	XMCGRA XMCGRA
XMCGRM	$\partial^2 M_{CG}/\partial R \partial m$	M	See symbol	/DYNA /(	209)	DL2 UT	I M	XMCGRM XMCGRM
XMCGRR	$\partial^2 M_{CG}/\partial \alpha^2$	M	See symbol	/DYNA /(	208)	DL2 UT	I M	XMCGRR XMCGRR
XMCGV	$\partial M_{CG}/\partial V$	M	See symbol	/DYNA /(	201)	DL2 UT	I M	XMCGV XMCGV

1504

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
XMCGVA	$\partial^2 M_{CG}/\partial V \partial \alpha$	M	See symbol		/DYNA /(	207)	DL2 UT	I M	XMCGVA XMCGVA
XMCGVM	$\partial^2 M_{CG}/\partial V \partial m$	M	See symbol		/DYNA /(	206)	DL2 UT	I M	XMCGVM XMCGVM
XMCGVR	$\partial^2 M_{CG}/\partial V \partial R$	M	See symbol		/DYNA /(	205)	DL2 UT	I M	XMCGVR XMCGVR
XMCGVV	$\partial^2 M_{CG}/\partial V^2$	M	See symbol		/DYNA /(	204)	DL2 UT	I M	XMCGVV XMCGVV
XT	X_T	I	Aerodynamic trim surface body x station		/ARCDAT/(	36)	UT	I	XT
ZCG	Z_{CG}	I	Center of gravity body z station	(FT)	/DYNA /(	158)	DL2 STATEF UT	I I I	ZCG ZCG ZCG
ZCGM	$\partial Z_{CG}/\partial m$	I	See symbol		/DYNA /(	110)	DL2 STATEF UT	I M I	ZCGM ZCGM ZCGM
ZCGMM	$\partial^2 Z_{CG}/\partial m^2$	I	See symbol		/DYNA /(	111)	DL2 STATEF UT	I M I	ZCGMM ZCGMM ZCGMM
ZCGR	Z_{CGR}	I	Reference zcg location	(FT)	/ARCDAT/(	33)	UT	I	ZCGR
ZE	Z_E	I	Engine thrust centroid body z station		/ARCDAT/(	35)	DL2 UT	I I	ZE ZE

1305

SUBRØUTINE WRAPUP

Purpose

WRAPUP controls the integration of the converged QL solution. It also writes that solution or logical unit 11 for the possible use later as a starting guess for steepest descent or QL itself. Finally, WRAPUP controls the QL trajectory to sizing interface.

```
      PROGRAM WRAPUP                                                    AUG10
C                                                                       JUL21
C                THIS ROUTINE CONTROLS THE INTEGRATION OF THE CON-      JUL21
C                VERGED QL SOLUTION.  IT ALSO WRITES THAT SOLUTION      JUL21
C                ON UNIT 11 FOR THE POSSIBLE USE LATER AS A STARTING    JUL21
C                SOLUTION FOR STEEP. DESC. OR QL ITSELF.  FINALLY,      JUL21
C                THIS ROUTINE CONTROLS THE QL TRAJ. TO SIZING INTER-    JUL21
C                FACE.                                                  JUL21
C                                                                       JUL21
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,WD                           SIZING
      COMMON /SIZING/                                                   SIZING
C         PHASE II SIZING PARAMERERS                                    SIZING
     *TZ,     VV(3),     QP(14),   EROR,    PZ(5),    VQ,      SW(20),  SIZING
     *SV(28), SQ(37,5),  SE(11),   TLAT,    TLNG,                       SIZING
C        PHASE I SIZING PARAMERERS                                      SIZING
     *WBO, -  WLOO,      DWEB,     DWEO,    TOLWT,    WPB,     TWRAT2,  SIZING
     *BK1,    BK2,       BK3,      BK4,     ISIZE,    TRAFLG,  TWRATO,  SIZING
     *OK1,    OK2,       OK3,      OK4,     PRFLG,    IPASS,   IPSMAX,  SIZING
     *AEXIT,  TVACO,     WO,       WFD,     IDVEL,    ISPO,    ISPB,    SIZING
     *XPL,    TVACB,   NNB,        WEO,     WEB,      WO,      WLO,     SIZING
     *DVO,     DVB,      MUB,       MUO,     VSTG,     WPO              SIZING
     *,JTYP,   BECO ,    BSTG ,    ORBI,    ITNBW , ITNOW  ,ISZO(23)    SIZING
      LOGICAL SWITCH, ILOAD                                             DYNA
      REAL MACH, ISP, ISPV, ISPR, ISPM, ISPT, ISPVV, ISPVR, ISPVM,      DYNA
     *ISPVT, ISPRR, ISPRM, ISPRT, ISPMM, ISPMT, ISPTT, LIFT, LIFTV,     DYNA
     *LIFTR, LIFTA, LIFTVV, LIFTVR, LIFTVA, LIFTRR, LIFTRA, MUR, LIFTAA, DYNA
     *IRATED, ISPF, ISPFF                                               DYNA
      REAL MACHV, MACHR, MACHVR, MACHRR                                 DYNA
      REAL LIFTM, LIFTVM, LIFTRM, LIFTMM, LIFTMA                        DYNA
      COMMON /DYNA/                                                     DYNA
     *XX    ,TIME  ,SINGAM,COSGAM,OMEGA ,OMEGA2,R   -  ,G     ,SINA-  , JUL21-
     *COSA  ,DYN011,OMEGAT,TAMP  ,PA    ,RO    ,CS    ,TEMPR ,PAR   ,   DYNA
     *ROR   ,CSR   ,TEMPRR,PARR  ,RORR  ,CSRR  ,KODE  ,MACH  ,Q     ,   DYNA
     *QV    ,QR    ,QVV   ,QVR   ,QRR   ,FVAC  ,FVACV ,FVACR ,FVACM ,   DYNA
     *FVACT ,FVACVV,FVACVR,FVACRR,FVACTT,T     ,MACHV ,MACHR ,ISP   ,   DYNA
     *ISPV  ,ISPR  ,ISPM  ,ISPT  ,ISPVV ,ISPVR ,ISPVM ,ISPVT ,ISPRR ,   DYNA
     *ISPRM ,ISPRT ,ISPMM ,ISPMT ,ISPTT ,LIFT  ,LIFTV ,LIFTR ,LIFTA ,   DYNA
     *LIFTVV,LIFTVR,LIFTVA,LIFTRR,LIFTRA,DRAG  ,DRAGV ,DRAGR ,DRAGA ,   DYNA
     *DRAGVV,DRAGVR,DRAGVA,DRAGRR,DRAGRA,DRAGAA,ALPHA ,PHI   ,LIFTM ,   DYNA
     *LIFTVM,LIFTRM,LIFTMM,LIFTMA,DBR   ,DBRR  ,GAMMAD,AE    ,TAX   ,   DYNA
     *W     ,SINPHI,COSPHI,SINPSI,COSPSI,SINRHO,COSRHO,SINROR,COSROR,   DYNA
     *MUR   ,XKG   ,XKP   ,AKIN  ,CDO   ,CDOM  ,CLO   ,FK    ,XCGM  ,   DYNA
     *XCGMM ,ZCGM  ,ZCGMM ,XJV   ,XJR   ,XJVV  ,XJVR  ,XJRR  ,MACHVR,   DYNA
     *MACHRR,SIN2RO,COS2RO,COS2GM,CM    ,CMA   ,CMM   ,CMAA  ,CMMM  ,   DYNA
     *CMAM  ,CMO   ,CMOM  ,CMOMM ,CMAMM ,ULFTV ,ULFTR ,ULFTVV,ULFTVR,   DYNA
     *ULFTVA,ULFTRR,ULFTRA,IPOW  ,XARC  ,TSTART,GH    ,GRR   ,LIFTAA,   DYNA
     *CDOMM ,CLAMM ,CLOM  ,CLOMM ,DYN149,CT    ,CODAE ,SIDAE ,COD   ,   DYNA
     *SID   ,DELTAE,COE   ,XCG   ,ZCG   ,XJ    ,XMCG  ,CALPHA,ALMAX ,   DYNA
     *DB    ,ULFT  ,CULFT ,ULFTA ,TSTAGE,TIMES ,XMCGAA,IRATED,FRATED    DYNA
      COMMON /DYNA/                                                     DYNA
     *MTT   ,J1    ,J2    ,J3    ,XMCGA ,FVACF ,ULFTAA,ISPF  ,ISPFF ,   DYNA
     *ILOAD ,FKM   ,FKMM  ,SWITCH,INOF  ,CL    ,CLA   ,CLM   ,CLAA  ,   DYNA
     *CLMM  ,CLAM  ,CD    ,CDA   ,CDM   ,CDAA  ,CDMM  ,CDAM  ,DYN198,   DYNA
     *DYN199,DYN200,XMCGV ,XMCGR ,XMCGM ,XMCGVV,XMCGVR,XMCGVM,XMCGVA,   DYNA
     *XMCGRR,XMCGRM,XMCGRA,XMCGMM,XMCGMA,RORRR ,DYN214,DYN215,DYN216,   DYNA
     *DYN217,IDAM  ,TAIRB ,TAIRBV,TAIRBH,TARBVV,TARBHH,TARBVH,SFC   ,   JUL21
     *SFCV  ,SFCH  ,SFCVV ,SFCHH ,SFCVH                                 AUG09
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF       ,EJ         ,XISP       ,TMULT      ,DTNC       ,DTPI       , ARCDAT
     *IATM       ,IMODE      ,JAER       ,JPRO       ,QMAX       ,GMAX       , ARCDAT
     *XLMAX      ,HDMAX      ,GMDOT      ,ALFMAX     ,PHMAX      ,MAEA       , ARCDAT
     *MAEB       ,MAEC       ,MAED       ,MAEE       ,MAEF       ,MAEG       , ARCDAT
     *MT         ,MISP       ,MXCG       ,MZCG       ,MWDA       ,MWDB       , ARCDAT
     *MDB        ,XCGR       ,ZCGR       ,XE         ,ZE         ,XT         , ARCDAT
     *DREF       ,MCND       ,RHOB       ,QMULT      ,REMAX      ,FRATE        ARCDAT
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      REAL MAGBV, MU, M, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU, NOM   D
     *, LHT                                                             D
      COMMON /D/                                                        D
     *X, H, XI(4), MAGBV, ERR, D9, D10, C(40), CSAVE(40), V, GAM, PSI,  D
     *ALT,RHO,MU,M, TAU, HT, LV, LGAM, LPSI, LR, LRHO, LMU, LM, LTAU,   JUL21
     *LHT, D109, D110, BV(40), ZSAVE(20), QT(20), NPOINT(20), DELT(20)  D
      DIMENSION NOM(20)                                                 D
      EQUIVALENCE (NOM, V)                                              D
```

```
      COMMON/GLOBAL/                                                    GLOBAL
     *GR     ,ER     ,OMGZ   ,XLAMRF,YMURF ,LUM   ,TO    ,EPSLON,INNER  GLOBAL
     *ITRMAX,JJOP(6)        ,IFATAL,NARC  ,NBRAN ,NFARC ,ID(4) ,KTAB(20),GLOBAL
     *ITAB(20), SIG,MAXTAB,GM,PSIRF ,IPFLG1,IPFLG2,IPFLG3,IPFLG4,       GLOBAL
     *INEQFL(20), ITPSO, KSOL, INARK,KGLOBL(7)                          GLOBAL
      COMMON /BLOCK/ IIC(10, 20), IICT(10, 20), ITC(10, 20), JTAB(20),  BLOCK
     *ITCT(10, 20), LTAB(20), NOKNOW, NOC(20), VALIC(10, 20),           BLOCK
     *VALTC(10, 20), IPAY                                               BLOCK
      COMMON /Y/ Y(820)                                                 Y
      COMMON /ZD/ ZD(50)                                                ZD
      COMMON /Z/ Z(50)                                                  Z
      COMMON /PC/                                                       PC
     *PC1    ,N      ,PC3   ,IDP   ,PC5   ,PC6   ,PC7   ,MAXBC ,NAUX    PC
      COMMON /CNTRL/                                                    CNTRL
     *NU     ,ITER   ,ITAPA ,ITAPB ,JMIN  ,JMAX  ,LINES ,KPT   ,MOM   ,  CNTRL
     *KARD   ,INDX(4)       ,NEWNOM,CNT016,RHOC  ,RHOP  ,NPTS  ,MINES ,  CNTRL
     *KPAGE  ,NNP    ,NUP   ,IARC  ,TRSTR ,IMAX  ,KTIME ,KONVER,NOPRNT,  CNTRL
     *INBDRY,NUPAGE,IVARY(20), NN, NOVARY, PLAST, ZLAST, KODES          CNTRL
      LOGICAL INBDRY, NEWNOM, KONVER, NOPRNT, NUPAGE                    CNTRL
      DATA WRAPUX/6HWRAPUP/                                             JUL21
    1 FORMAT(1H0, 35HTHE FOLLOWING ITERATE IS CONVERGED./)              JUL21
      EQUIVALENCE(TT, X), (DT, H)                                       JUL21
      DIMENSION S(820)                                                  JUL21
      DIMENSION ZERO(20)                                                JUL21
      DATA ZERO/20*0./                                                  JUL21
      DATA DEG/57.2957795130823/                                        JUL21
      CALL ETIME                                                        JUL21
C                                                                       JUL21
C     WRITE CONVERGENCE NOTE                                            JUL21
C                                                                       JUL21
      WRITE(6, 1)                                                       JUL21
C                                                                       JUL21
C     COMBINE UP THE LAST NOMINAL.                                      JUL21
C                                                                       JUL21
      L = 1 - N                                                         JUL21
      MOM = 0                                                           JUL21
      DO 70 IARC = 1, NARC                                              JUL21
      L = L + N                                                         JUL21
      MOM = NOC(IARC)                                                   JUL21
      NN = N*(MOM + 1)                                                  JUL21
      CALL READMS(41, S, NN, 2*IARC - 1)                                JUL21
      CALL MATMLT(Y(L), S(N + 1), C, N, MOM, 1)                         JUL21
      CALL MATADD(Y(L), Y(L), S, N, 1)                                  JUL21
      IF(IARC - 1) 50, 50, 60                                           JUL21
   50 QT(IARC) = Y(L + 7) + TO                                          JUL21
      GO TO 70                                                          JUL21
   60 QT(IARC) = Y(L + 7) + QT(IARC - 1)                                JUL21
   70 CONTINUE                                                          JUL21
      REWIND INARK                                                      JUL21
      NS = N/2 - 1                                                      JUL21
      WRITE(INARK) TO, NS, NARC, (QT(IK), IK = 1, NARC), (ZERO(IK), IK = AUG22
     *1, NARC), (ZERO(IK), IK = 1, NARC), (ZERO(IK), IK = 1, NARC)      JUL21
C                                                                       JUL21
      DO 71 I = 1, 50                                                   JUL21
   71 Z(I) = 0.                                                         JUL21
      NN = N + NAUX                                                     JUL21
      CALL QINTRI                                                       JUL21
      L = -N                                                            JUL21
      KODES = -1                                                        JUL21
      IDAM = 1                                                          JUL21
C                                                                       JUL21
C     START THE INTEGRATION.                                            JUL21
C                                                                       JUL21
   81 DO 13 IARC = 1, NARC                                              JUL21
C                                                                       JUL21
C     STORE FIRST POINT OF SUBARC IN Z-ARRAY.                           JUL21
C                                                                       JUL21
      L = L + N                                                         JUL21
      DO 9 I = 1, N                                                     JUL21
      J = L + I                                                         JUL21
    9 Z(I) = Y(J)                                                       JUL21
C                                                                       JUL21
C     STORE INITIAL POINT OF SUBARC.                                    JUL21
C                                                                       JUL21
```

50 60

70

```
      TI = IARC - 1                                                     JUL21
C                                                                       JUL21
C     COMPUTE FINAL POINT OF SUBARC.                                    JUL21
C                                                                       JUL21
   10 TF = IARC                                                         JUL21
C                                                                       JUL21
C     READ IN THE DATA FOR THIS SUBARC.                                 JUL21
C                                                                       JUL21
      CALL READMS(9, ARCDA, 42, IARC)                                   JUL21
C                                                                       JUL21
C     COMPUTE STEPSIZE.                                                 JUL21
C                                                                       JUL21
      DT = DTNC/Z(IDP)                                                  JUL21
      NPTS = 1000000                                                    JUL21
      NEWNOM = .TRUE.                                                   JUL21
      TT = TI                                                           JUL21
      TP = TI                                                           JUL21
      KPT = 0                                                           JUL21
   12 KPT = KPT + 1                                                     JUL21
      CALL NLDRV(Z, ZD)                                                 JUL21
      TS = Z(IDP)*(TT - TI)                                             JUL21
      CALL ENVPRO                                                       JUL21
      ALFDEG = ALPHA*DEG                                                JUL21
      PHIDEG = PHI*DEG                                                  JUL21
      WRITE(INARK) IARC,TIME,IS,ALFDEG,PHIDEG,V,GAM,ALT,M,PSI,RHO,MU,HT, JUL21
     *LV,LGAM,LR,LM,LPSI,LRHO,LMU,LHT,LTAU                              JUL21
      IF(ABS(TP -TT) .GE. 1.E-8) GO TO 120                              JUL21
      CALL OUTPUT                                                       JUL21
      TP = TP + DTPI*DT                                                 JUL21
      IF(ABS(TF - TT) .LT. 1.E-8) GO TO 125                             JUL21
  120 IF(TT+ DT .LT. TF - 1.E-8) GO TO 121                              JUL21
      DT = TF - TT                                                      JUL21
      NPTS = KPT + 1                                                    JUL21
      TP = TF                                                           JUL21
  121 CALL RKUTT2                                                       JUL21
      GO TO 12                                                          JUL21
  125 IF(IARC.EQ.NARC) GO TO 13                                         JUL21
      IF(JTYP .GT. 0)  CALL QLAEND                                      JUL21
   13 CONTINUE                                                          JUL21
      ENDFILE INARK                                                     JUL21
      REWIND INARK                                                      JUL21
      CALL TRJNDQ                                                       JUL21
      RETURN                                                            JUL21
      END                                                               JUL21
```

FORTRAN SYMBOL	MATH SYMBOL[1]	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALPHA	α	I	Angle of attack	(RAD)	/DYNA /(	79)	AEROCO ALGCON AL2 ARCIN CONTRL ENVPRO MOMECO NPLANE OUTPUT TRAJIN UT WRAPUP	I M I M M I I I I O I I	ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA ALPHA
ALT	h	I	Altitude	(FT)	/D /(	94)	OUTPUT STATEF WRAPUP	I I I	ALT ALT ALT
ARCDA	S_{ref}	I	Aerodynamic reference area	(FT^2)	/ARCDAT/(	1)	ARCIN BNDRY CHECK FETCH SALVE STATEF UT WRAPUP	I I I I I I I I	SREF ARCDA ARCDA ARCDA ARCDA SREF SREF ARCDA
C	c	I	A forty word array containing the vector of c's, i.e. the multipliers for the homogeneous solutions.		/D /(	11)	BNDRY BRANPT GROPE INTRPT NEWCS NLDRV NOMNAL WRAPUP	I I I I M I I I	C C C C C C C C
DT	h	M	Integration step size in quasitime.		/D /(	2)	AL4 INARC MADAMS RKUTT1 RKUTT2 SALVE WRAPUP	I M I I I M M	H H H H H H DT
DTNC	$\Delta\tau$	I	Integration interval	(SEC)	/ARCDAT/(	5)	WRAPUP	I	DTNC
DTPI		I	Print frequency for trajectory		/ARCDAT/(	6)	WRAPUP	I	DTPI
GAM	γ	I	Relative flight path angle.	(RAD)	/D /(	92)	ARCIN ENVPRO OUTPUT STATEF WRAPUP	I I I I I	GAM GAM GAM GAM GAM
HT	Q	I	Heating	(BTU)	/D /(	99)	OUTPUT WRAPUP	I I	HT HT
IARC	I	M	Subarc number.		/CNTRL /(	24)	ARCIN BCOND BNDRY BRANPT CHECK COSTAB COSTAI ENDPT FORCES INARC INTRPT MAGIC MARCH QLTOSZ SALVE WRAPUP	I M M I M I I I I M I M I I M M	IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC IARC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
IDAM		O	Optional atmospheric calculations flag. IDAM = -1: Compute $\partial^3\rho_a/\partial R^3$; IDAM = 0: No optional calculations; IDAM = 1: Compute $\partial^3\rho_a/\partial R^3$, μ_a, $\partial\mu_a/\partial R$, etc.	/DYNA /(	218)	ARCIN	O	IDAM
						ERROR	I	IDAM
						NPLANE	O	IDAM
						STATEF	I	IDAM
						WRAPUP	O	IDAM
IDP		I	Component number that corresponds to the OL state variable τ IDP = 8.	/PC /(	4)	INARC	I	IDP
						WRAPUP	I	IDP
INARK		I	Logical unit on which initial and converged arcs are stored. INARK = 11.	/GLOBAL/(	95)	CHECK	O	INARK
						FETCH	I	INARK
						INARC	I	INARK
						MARCH	I	INARK
						WRAPUP	I	INARK
KODES		O	Not used.	/CNTRL /(	56)	GROPE	O	KODES
						NLDRV	M	KODES
						WRAPUP	O	KODES
KPT		M	The subarc point number. KPT = 1 on the first point of subarc, and KPT = NPTS on the last point of the subarc.	/CNTRL /(	8)	BCOND	O	KPT
						BNDRY	O	KPT
						FORCES	I	KPT
						MAGIC	O	KPT
						RKUTT1	I	KPT
						SALVE	M	KPT
						WRAPUP	M	KPT
LGAM	λ_γ	I	Relative flight path angle costate	/D /(	101)	ALI	I	LGAM
						ARCIN	I	LGAM
						CONTRL	I	LGAM
						NLDRV	I	LGAM
						OUTPUT	I	LGAM
						WRAPUP	I	LGAM
LHT	λ_Q	I	Heating costate	/D /(	108)	NLDRV	I	LHT
						WRAPUP	I	LHT
LM	λ_m	I	Mass costate	/D /(	106)	NLDRV	I	LM
						OUTPUT	I	LM
						WRAPUP	I	LM
LMU	λ_μ	I	Relative longitude costate	/D /(	105)	NLDRV	I	LMU
						OUTPUT	I	LMU
						WRAPUP	I	LMU
LPSI	λ_ψ	I	Relative azimuth angle costate	/D /(	102)	ALI	I	LPSI
						ARCIN	I	LPSI
						CONTRL	I	LPSI
						NLDRV	I	LPSI
						OUTPUT	I	LPSI
						WRAPUP	I	LPSI
LR	λ_R	I	Altitude costate	/D /(	103)	NLDRV	I	LR
						OUTPUT	I	LR
						WRAPUP	I	LR
LRHO	λ_ρ	I	Latitude costate	/D /(	104)	NLDRV	I	LRHO
						OUTPUT	I	LRHO
						WRAPUP	I	LRHO
LTAU	λ_τ	I	Subarc duration costate	/D /(	107)	OUTPUT	I	LTAU
						WRAPUP	I	LTAU
LV	λ_v	I	Relative velocity costate	/D /(	100)	ALI	I	LV
						CONTRL	I	LV
						NLDRV	I	LV
						OUTPUT	I	LV
						WRAPUP	I	LV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
M	m	I	Mass	(G'S)	/D /(	97)	AL4	I	M
							AL7	I	M
							AL8	I	M
							AL9	I	M
							APPLY	I	M
							BRANPT	I	M
							COSTAB	I	M
							COSTAI	I	M
							INTRPT	I	M
							NLDRV	I	M
							OUTPUT	I	M
							SALVE	I	M
							STATEF	I	M
							WRAPUP	I	M
MOM		M	*The number of homogeneous solutions currently being integrated.*		/CNTRL /(	9)	GROPE	O	MOM
							INARC	M	MOM
							LINDRV	I	MOM
							NOMNAL	I	MOM
							SALVE	M	MOM
							WRAPUP	M	MOM
MU	μ	I	Relative longitude	(RAD)	/D /(	96)	OUTPUT	I	MU
							PDBCOL	I	MU
							WRAPUP	I	MU
N		I	Total number of QL state and costate variables. N = 18.		/PC /(	2)	BNDRY	I	N
							CHECK	I	N
							INARC	I	N
							LINDRV	I	N
							NLDRV	I	N
							NOMNAL	I	N
							RKUTT1	I	N
							SALVE	I	N
							WRAPUP	I	N
NARC	N_3	I	Number of subarcs in the problem.		/GLOBAL/(	18)	BCOND	I	NARC
							BNDRY	I	NARC
							CHECK	I	NARC
							ENDPT	I	NARC
							ENVPRQ	I	NARC
							FETCH	I	NARC
							INARC	I	NARC
							MAGIC	I	NARC
							QLTOSZ	I	NARC
							SALVE	I	NARC
							WRAPUP	I	NARC
NAUX		I	Number of velocity loss quantities to be integrated on converged trajectory. NAUX = 5.		/PC /(	9)	WRAPUP	I	NAUX
NEWNOM		O	A logical flag that indicates to the Runge-Kutta integration whether or not the system Jacobian needs to be reevaluated.		/CNTRL /(	15)	INTERP	O	NEWNOM
							LINDRV	M	NEWNOM
							RKUTT1	O	NEWNOM
							SALVE	O	NEWNOM
							WRAPUP	O	NEWNOM
NN		M	The number of quantities currently being numerically integrated.		/CNTRL /(	52)	BNDRY	M	NN
							INARC	M	NN
							MADAMS	I	NN
							MAGIC	M	NN
							NOMNAL	I	NN
							RKUTT1	I	NN
							RKUTT2	I	NN
							SALVE	M	NN
							WRAPUP	M	NN
NOC		I	An array containing a running total of the number of free (unknown) state and costate variables at the start of each subarc.		/BLOCK /(	842)	BNDRY	I	NOC
							BRANPT	I	NOC
							COSTAB	O	NOC
							COSTAI	O	NOC
							COSTAO	O	NOC
							INARC	I	NOC
							INTRPT	I	NOC
							SALVE	I	NOC
							WRAPUP	I	NOC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
NPTS		O	The total number of points in the subarc.		/CNTRL	/(19)	BCOND	O	NPTS
							BNDRY	O	NPTS
							FORCES	I	NPTS
							INARC	M	NPTS
							MAGIC	O	NPTS
							SALVE	M	NPTS
							WRAPUP	O	NPTS
PHI	ϕ	I	Bank angle	(RAD)	/DYNA	/(80)	CONTRL	M	PHI
							OUTPUT	I	PHI
							WRAPUP	I	PHI
PSI	ψ	I	Relative azimuth angle.	(RAD)	/D	/(93)	OUTPUT	I	PSI
							STATEF	I	PSI
							WRAPUP	I	PSI
QT		M	A twenty word array containing the values from the initial arc of the successive subarcs' durations		/D	/(171)	INARC	O	QT
							WRAPUP	M	QT
RHO	ρ	I	Latitude	(RAD)	/D	/(95)	AL9	I	RHO
							OUTPUT	I	RHO
							STATEF	I	RHO
							WRAPUP	I	RHO
TIME		I	Trajectory time	(SEC)	/DYNA	/(2)	ENVPRO	I	TIME
							OUTPUT	I	TIME
							PDBCOL	I	TIME
							STATEF	M	TIME
							WRAPUP	I	TIME
TT	x	M	The quasitime variable.		/D	/(1)	AL4	I	X
							BNDRY	O	X
							ERROR	I	X
							FETCH	O	X
							FORCES	I	X
							INARC	M	X
							INTERP	I	X
							MADAMS	M	X
							RKUTT1	M	X
							RKUTT2	M	X
							SALVE	M	X
							STATEF	I	X
							WRAPUP	M	TT
TO	t_0	I	Trajectory start time.	(SEC)	/GLOBAL	/(7)	FETCH	M	TO
							INARC	M	TO
							TRAJIN	I	TO
							WRAPUP	I	TO
V	V	I	Relative velocity.	(FT/SEC)	/D	/(91)	AL1	I	V
							AL4	I	V
							AL7	I	V
							AL8	I	V
							AL9	I	V
							BCOND	I	NOM
							BNDRY	O	NOM
							BRANPT	M	NOM
							CONTRL	I	V
							ENDPT	I	NOM
							ENVPRO	I	V
							FETCH	O	NOM
							INTERP	M	V
							INTRPT	M	NOM
							NLDRV	O	NOM
							NLDRV	I	V
							OUTPUT	I	V
							PDBCOL	I	V
							STATEF	I	V
							WRAPUP	I	V
Y		I	An 820 word array containing the particular and homogeneous solutions being integrated. The first 18 words comprise the particular solution. Each block of 18 words thereafter comprises an independent homogeneous solution.		/Y	/(1)	GROPE	O	Y
							INARC	M	Y
							MADAMS	M	Y
							QLTOSZ	I	Y
							RKUTT1	M	Y
							SALVE	M	Y
							WRAPUP	I	Y

1314

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
Z	Z	M	A 20 word array used to store the total linear solution from the preceding QL iteration.	/Z	/(1)	BNDRY	I	Z
						BRANPT	I	Z
						ENDPT	I	Z
						ENVPRO	I	Z
						INTERP	O	Z
						INTRPT	I	Z
						LINDRV	I	Z
						NOMNAL	M	Z
						OUTPUT	I	Z
						RKUTT1	O	Z
						RKUTT2	M	Z
						SALVE	M	Z
						WRAPUP	M	Z
ZD		I	A 20 word array containing the vector f(X,Z,W) in Equation 17.1-7 in Vol.I of this document.	/ZD	/(1)	ENVPRO	I	ZD
						LINDRV	I	ZD
						OUTPUT	I	ZD
						RKUTT2	I	ZD
						WRAPUP	I	ZD

1315

SUBRØUTINE SIZIN

1316

SIZIN

Purpose

SIZIN is the routine that adjusts trajectory data for sizing solutions.

Description

SIZIN makes necessary changes in several types of trajectory data after each pass through either in the phase I sizing or in the SSSP module.

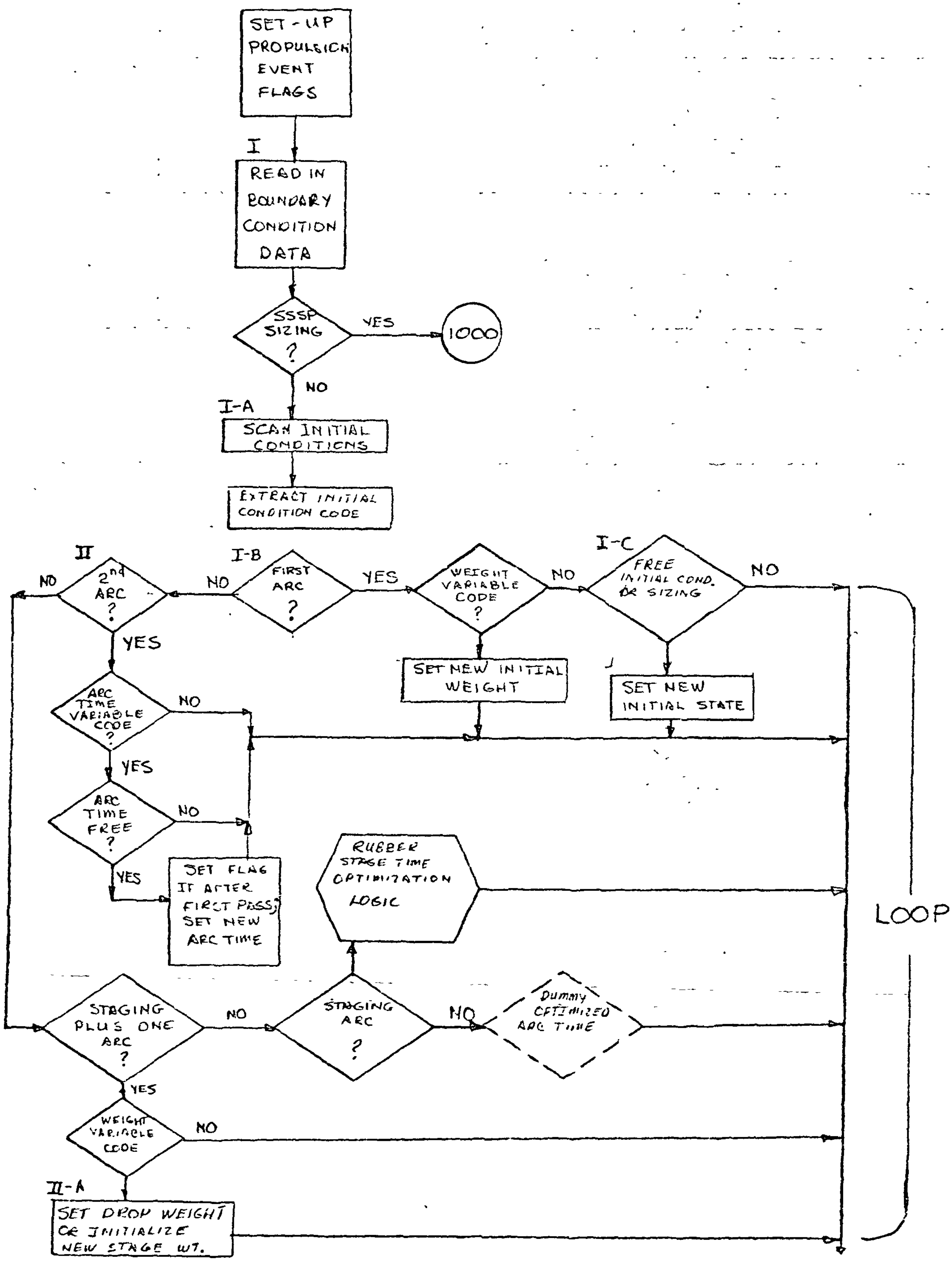
1. Boundary Condition Data

   Vehicle initial weight, cut-off, and drop weights as well as optimized initial states are reset by SIZIN. This is done by scanning for and changing data contained on the boundary condition random access file. This is file 9, record 21.

2. Arc Data

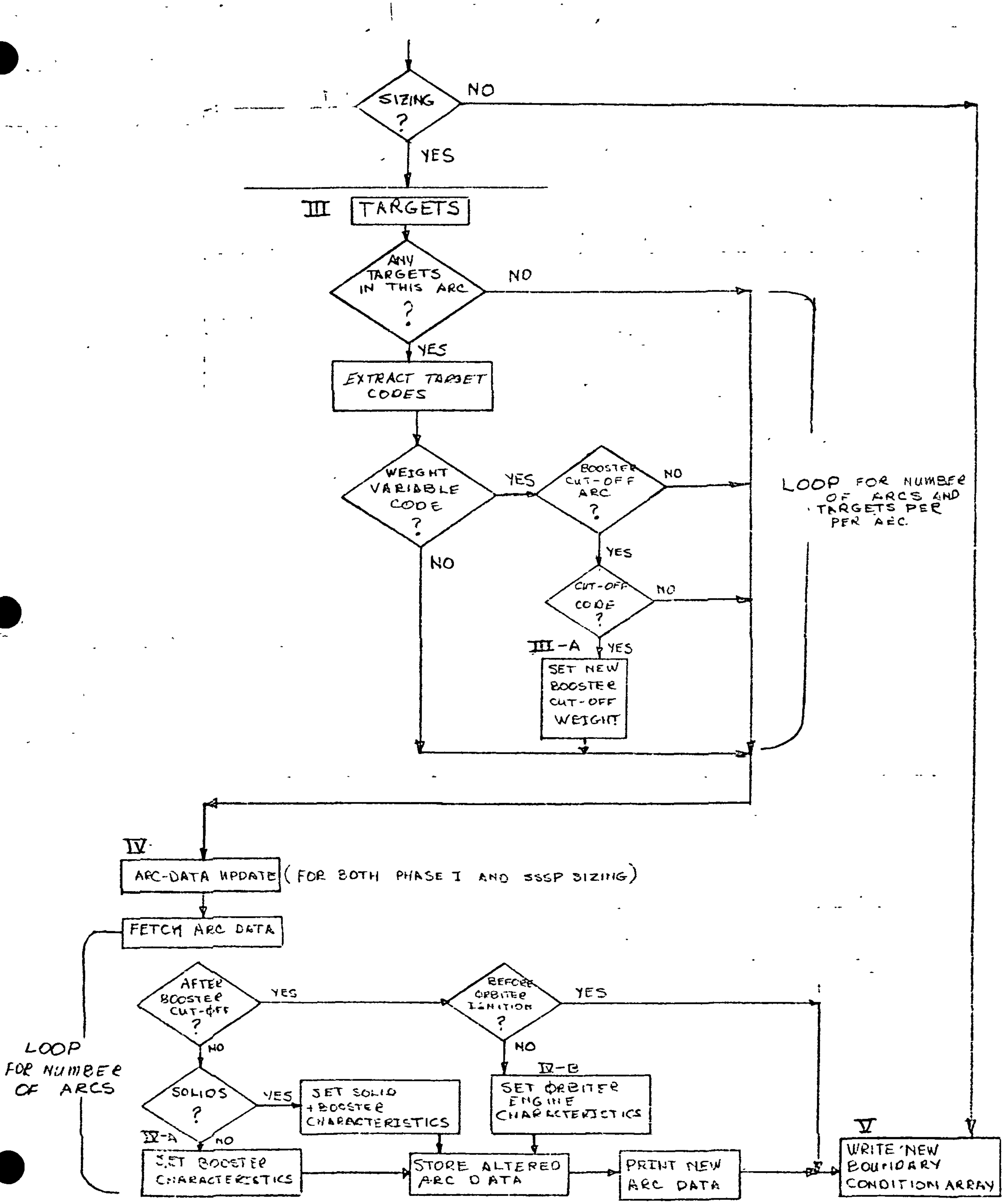
   Propulsion system and aerodynamic reference areas are modified according to changes generated during sizing, and the modified data is placed back on the arc data random file 9, records 1 through 20.

SIZIN



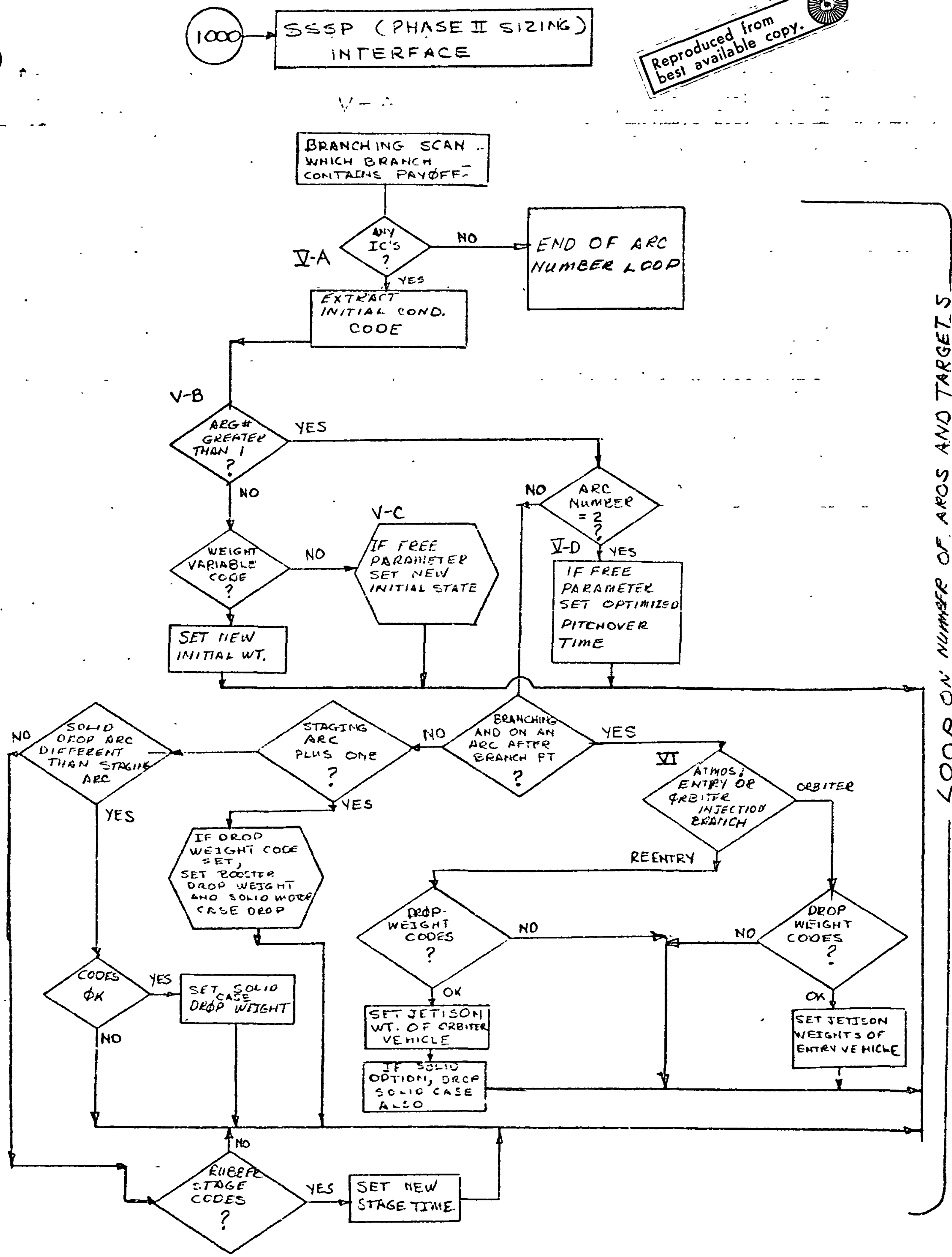
SET-UP PROPULSION EVENT FLAGS
I
READ IN BOUNDARY CONDITION DATA
SSSP SIZING ?
YES
1000
NO
I-A
SCAN INITIAL CONDITIONS
EXTRACT INITIAL CONDITION CODE
II
2nd ARC ?
NO
I-B
FIRST ARC ?
NO
YES
WEIGHT VARIABLE CODE ?
NO
I-C
FREE INITIAL COND. OR SIZING
NO
YES
SET NEW INITIAL WEIGHT
SET NEW INITIAL STATE
ARC TIME VARIABLE CODE ?
NO
YES
ARC TIME FREE ?
NO
YES
SET FLAG IF AFTER FIRST PASS; SET NEW ARC TIME
RUBBER STAGE TIME OPTIMIZATION LOGIC
LOOP
STAGING PLUS ONE ARC ?
NO
STAGING ARC ?
NO
Dummy OPTIMIZED ARC TIME
YES
WEIGHT VARIABLE CODE
NO
II-A
SET DROP WEIGHT OR INITIALIZE NEW STAGE WT.

SIZIN (CONTINUED)



1319

SIZIN (CONTINUED)



SIZIN (CONTINUED)

TARGET SCAN
FOR SSCP SIZING

SCAN TARGETS
TO SET
BOOSTER
CUT-OFF WEIGHT
(SEE III)

V

/32/

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ARCDA	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	BNTG	I	ARCDA
						EQUA3	I	SREF
						FNTG	I	ARCDA
						FXDAT	I	ARCDA
						FXDAT	O	IARCDA
						GEINP	M	ARCDA
						SDINP	I	ARCDA
						SIZIN	I	ARCDA
						SIZIN	M	SREF
						THRUST	I	SREF
						VT	I	SREF
EJ	A_{exit}	O	Nozzle exit area (FT^2)	/ARCDAT/(	2)	PROPB	I	EJ
						PROPIN	I	EJ
						SIZIN	O	EJ
FRATE		O	Input rated vacuum thrust per engine (LBS)	/ARCDAT/(	42)	EQUA3	I	FRATE
						FXDAT	I	FRATE
						PROPB	I	FRATE
						PROPIN	I	FRATE
						SIZIN	O	FRATE
IPASS		I	Sizing iteration counter	/SIZING/(	291)	GEINP	O	IPASS
						PADS1	M	IPASS
						PAVO2	I	IPASS
						SIZE	M	IPASS
						SIZIN	I	IPASS
						SSSP	M	IPASS
ITAB		I	A 20 word array containing the number of nonzero state initial conditions specified at the beginning of each subarc.	/GLOBAL/(	45)	GEINP	I	ITAB
						SDINP	I	ITAB
						SIZIN	I	ITAB
JTYP		I	Sizing. Flag.	/SIZING/(	313)	FNTG	I	JTYP
						GEINP	O	JTYP
						MODELA	I	JTYP
						PADS1	I	JTYP
						PROPIN	I	JTYP
						SIZIN	I	JTYP
						TRTOSZ	I	JTYP
KTAB		I	A 20 word array containing the number of state target conditions specified at the end of each subarc.	/GLOBAL/(	25)	GEINP	I	KTAB
						SDINP	I	KTAB
						SIZIN	I	KTAB
NARC	N_S	I	Number of subarcs in the problem.	/GLOBAL/(	18)	FNTG	I	NARC
						GEINP	M	NARC
						PROPIN	I	NARC
						SDINP	I	NARC
						SIZIN	I	NARC
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						POBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHOF	M	SQ
						WTVOL	M	SQ

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION		STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SREF	S_{ref}	M	Aerodynamic reference area	(FT^2)	/ARCDAT/(	1)	BNTG	I	ARCDA
							EQUA3	I	SREF
							FNTG	I	ARCDA
							FXDAT	I	ARCDA
							FXDAT	O	IARCDA
							GEINP	M	ARCDA
							SDINP	I	ARCDA
							SIZIN	I	ARCDA
							SIZIN	M	SREF
							THRUST	I	SREF
							VT	I	SREF
SV		I	A synthesis array (28) containing staging parameters and misc flags		/SIZING/(	46)	ENVPRM	M	SV
							FLYBKP	I	SV
							ITER8	I	SV
							RANGE	I	SV
							SIZEMR	M	SV
							SIZIN	I	SV
							SSSP	I	SV
							SUMOUT	I	SV
							TAMPAR	O	SV
							TAMPER	M	SV
							TRTOSZ	M	SV
							VEHDF	M	SV
							WTVOL	I	SV
TMULT	T_{mult}	O	Thrust multiplier or number of engines		/ARCDAT/(	4)	EQUA3	I	TMULT
							FXDAT	M	TMULT
							PROPB	I	TMULT
							PROPIN	I	TMULT
							SIZIN	O	TMULT
XISP	I_{sp}	O	Vacuum specific impulse	(SEC)	/ARCDAT/(	3)	IMPUL	I	XISP
							SIZIN	O	XISP

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE		SUBROUTINE USAGE		
				BLOCK	LOC	SUBR	CODE	VAR
.UN06.		0	File of all output data	/.UN06./($	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAY02	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHOF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      PROGRAM  SIZIN                                                    SIZIN
C      THIS  PROGRAM ADJUSTS  TRAJECTORY  DATA  FOR  SIZING  RUNS       SIZIN
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),      QP(14),    EROR,     PZ(5),     VQ,     SW(20), SIZING
     *SV(28),  SQ(37,5),   SE(11),    TLAT,     TLNG,                   SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,      DWEB,      DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,     NNB,       WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,     DVB,       MUB,       MUO,      VSTG,      WPO              SIZING
     *,JTYP,   BECO,      BSTG,      ORBI,     ITNBW,  ITNOW  ,            SIZING
     * SVDPSQ  , SVDCON  ,IHUNT    ,IOPSTG   ,ISZD(19)                  UN
      COMMON/GLOBAL/                                                    GLOBAL
     *GR        ,ER        ,OMGZ      ,XLAMRF    ,VRURF     ,LUM        GLOBAL
     *,JJQP(10) ,IFATAL    ,NARC      ,NBRAN     ,NFARC     ,ID(4)      GLOBAL
     *,KTAB(20),ITAB(20)   ,SIG       ,MAXTAB                           GLOBAL
     *,GM       ,PSIRF,IPFLG1,        IPFLG2,IPFLG3,IPFLG4,INEQFL(20)   GLOBAL
     *,ITPSO  ,KSOL       ,KGLOBL(8)                                    RETAP
      DIMENSION  BNARR(400),TARG(100)                                   SIZIN
      EQUIVALENCE(BNARR(301),TARG)                                      SIZIN
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF      ,EJ        ,XISP      ,TMULT     ,DTNC      ,DTPI     , ARCDAT
     *IATM      ,IMODE     ,JAER      ,JPRO      ,QMAX      ,GMAX     , ARCDAT
     *XLMAX     ,HDMAX     ,GMDOT     ,ALFMAX    ,PHMAX     ,MAEA     , ARCDAT
     *MAEB      ,MAEC      ,MAED      ,MAEE      ,MAEF      ,MAEG     , ARCDAT
     *MT        ,MISP      ,MXCG      ,MZCG      ,MWDA      ,MWDB     , ARCDAT
     *MDB       ,XCGR      ,ZCGR      ,XE        ,ZE        ,XT       , ARCDAT
     *DREF      ,MCND      ,RHOB      ,QMULT     ,REMAX                 ARCDAT
     * ,FRATE   , ARCD(9)                                               RETAP
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      IF(JTYP.EQ.0) GO TO 10                                            SIZIN
      IBECO = SQ(1,1)                                                   SIZIN
      IBESP = SQ(1,2)                                                   SIZIN
      ISOCO = SQ(1,4)                                                   SIZIN
      ISOSP = SQ(1,5)                                                   SIZIN
      IORBI = SQ(1,3)                                                   SIZIN
      IF(JTYP.NE.2) IOPSTG = 0                                          OS
C      I    READ IN  BOUNDARY CONDITIONS                                SIZIN
   10 CALL READMS(9,BNARR,400,21)                                       SIZIN
      IF(JTYP.EQ.2) GO TO  1000                                         SIZIN
C      I-A  SCAN INITIAL CONDITION                                      SIZIN
      KI= 0                                                             SIZIN
      DO 300IARC=1,NARC                                                 SIZIN
      KK= ITAB(IARC)                                                    SIZIN
      IF(KK.EQ.0) GO TO 300                                             SIZIN
      DO 100J=1,KK                                                      SIZIN
      ICD = BNARR(KI+1)                                                 SIZIN
      IVD = BNARR(KI+2)                                                 SIZIN
      YZ  = BNARR(KI+3)                                                 SIZIN
C  I-B TEST  FOR  FIRST ARC                                             SIZIN
      IF(IARC.GT.1) GO TO 200                                           SIZIN
C                                                                       SIZIN
      IF(IVD.NE.5) GO TO  20                                            SIZIN
C       I- B    SET   NEW  INITAL  WEIGHT                               SIZIN
      IF(ICD .NE.2.AND.JTYP .EQ.0)  GO TO  90                           SIZIN
      BNARR(KI+3) =VV(1)                                                SIZIN
      GO TO 90                                                          SIZIN
C       I-C   SET  NEW  INITIAL  STATES                                 SIZIN
   20 IF(ICD.NE.2.OR.IPASS.LE.2) GO TO 90                               SIZIN
      GO TO (   90,21,21,21,22,22,22),IVD                               SIZIN
   21 BNARR(KI+3) =   SQ(17,IVD-1)                                      SIZIN
      GO TO   90                                                        SIZIN
   22 BNARR(KI+3) = SQ(17,IVD -2)                                       SIZIN
      GO  TO   90                                                       SIZIN
C       II    ARCS AFTER FIRST                                          SIZIN
  200 IF(IARC.NE.2)  GO TO 210                                          SIZIN
      IF(IVD.NE.1)   GO TO 90                                           SIZIN
      IF(ICD.NE.2)   GO TO 90                                           SIZIN
```

```
      SQ(13,3)=2.                                                       SIZIN
      IF(IPASS.LE.2) GO TO 90                                           SIZIN   90
      BNARR(KI+3) =  SQ(18,3)                                           SIZIN
      GO  TO  90                                                        SIZIN   90
  210 IF(IARC.NE.IBESP  +1) GO TO 220                                   SIZIN  220
      IF(IVD.NE.5)   GO   TO    90                                      SIZIN   90
      IF(ICD.EQ.1) GO TO  212                                           SIZIN  212
C       II-A     DROP  WEIGHT  SET     ICD=5 OR 6                       SIZIN
      IF(ICD.LT.5.OR.ICD.GT.6) GO TO  90                                SIZIN   90
      BNARR(KI+3) = QP(9) - SV(7)                                       SIZIN
      GO TO  90                                                         SIZIN   90
C       II-B   WEIGHT  AT  BEGINNING  NEXT  ARC IS KNOWN                SIZIN
  212 BNARR(KI+3) =  SV(7)                                              SIZIN
      GO TO  90                                                         SIZIN   90
C     II-C  OPTIMAL  BOOST  TIME  DURATION                              SIZIN
  220 IF(IARC.NE.IBESP)  GO TO 230                                      SIZIN  230
      IF(IVD.NE.1)  GO TO 90                                            SIZIN   90
      IF(ICD.NE.2)  GO TO 90                                            SIZIN   90
      IF(IPASS.LE.2) GO TO 1190                                         AAA   1190
      BNARR(KI+3)=SQ(18,2)                                              UH
      GO TO  90                                                         SIZIN   90
  230 IF(IVD NE.1)  GO TO 90                                            SIZIN   90
      IF(ICD.NE.2)  GO TO 90                                            SIZIN   90
      IF(IPASS.LE.2) GO TO 90                                           SIZIN   90
C      INSERT  SPECIAL  CODING TO  SET  OPTIMAL ARC TIMES               SIZIN
   90 KI=KI + 3                                                         SIZIN
  100 CONTINUE                                                          SIZIN
  300 CONTINUE                                                          SIZIN
C           TEST  FOR  NO  SIZING                                       SIZIN
      IF(JTYP.EQ.0) GO TO 800                                           SIZIN  800
C      III     SCAN  TARGET  CONDITIONS                                 SIZIN
      KT= 0                                                             SIZIN
      DO 400 I=1,NARC                                                   SIZIN
      KL= KTAB(I)                                                       SIZIN
      IF(KL.EQ.0) GO TO 400                                             SIZIN  400
      DO 390 J=1,KL                                                     SIZIN
      ICD = ICOD(TARG(KT+1))                                            SIZIN
      IVD = IVOD(TARG(KT+1))                                            SIZIN
      IVD=IABS(IVD)                                                     SIZIN
      VZ  = TARG(KT+2)                                                  SIZIN
      LCD =IABS(ICD)                                                    SIZIN
      IF(IVD.NE.5) GO   TO   380                                        SIZIN  380
      IF(I   .NE.IBECO) GO TO 380                                       SIZIN  380
      IF(LCD.NE.1)  GO TO   380                                         SIZIN  380
C      III-A     BOOSTER  CUT-OFF  WEIGHT                               SIZIN
      TARG(KT+2) = QP(9)                                                SIZIN
  380 KT= KT+2                                                          SIZIN
  390 CONTINUE                                                          SIZIN
  400 CONTINUE                                                          SIZIN
      IF(JTYP.EQ.0) GO TO 800                                           SIZIN  800
  401 CONTINUE                                                          AAA
C      IV      SCAN  AND  CORRECT   THRUST AND ISP                      SIZIN
      WRITE(6,450)                                                      SIZIN
  450 FORMAT(1H1,20X,27HARC DATA UPDATED BY SIZE      )                 SIZIN
      DO  600 IARC=1,NARC                                               SIZIN
      CALL  READMS( 9,SREF,51,IARC)                                     SIZIN
      IF(IARC.GT.IBECO) GO TO 510                                       SIZIN  510
C         TEST  FOR  SOLIDS                                             AAA
      IF(SQ(20,1).NE.0. AND.IARC.LE.ISOCO) GO TO 470                    AAA    470
C      IV-A    SET  ENGINE  CHARACTERISTICS   (BOOSTER)                 SIZIN
      FRATE= QP(1)                                                      SIZIN
      XISP=  QP(3)                                                      SIZIN
      TMULT=1.                                                          SIZIN
      IF(QP(7).NE.0) SREF=QP(7)                                         SIZIN
      EJ  = QP(11)                                                      SIZIN
      GO TO  580                                                        SIZIN   580
C         SOLID + BOOSTER  CHARACTERISTICS                              AAA
  470 FRAT = QP(1)+ SQ(22,4)                                            AAA
      XISP  = TZ                                                        AAA
      TMULT = 1.                                                        AAA
```

```
      EJ   =  SQ(22,5) +QP(11)                                          AAA
      GO  TO  580                                                       AAA
C       IV-B         ENGINE       ORBITER                               SIZIN
  510 IF(IARC.LT.IORBI) GO TO  600                                      SIZIN
      FRATE= QP(2)                                                      SIZIN
      XISP = QP(4)                                                      SIZIN
      TMULT=1.                                                          SIZIN
      IF(QP(8).NE.0) SREF =QP(8)                                        SIZIN
      IF(QP(12).NE.0.) EJ=QP(12)                                        UH
  580 CALL WRITMS( 9,SREF,51,IARC)                                      SIZIN
      N=IARC                                                            SIZIN
      WRITE(6,70) N,(ARCDA(I),I=1,17),(ARCDA(J),J=32,37) ,ARCDA(41),    SIZIN
     *ARCDA(42),                                                        SIZIN
     1(ARCDA(K),                                                        SIZIN
     2K=18,31),ARCDA(38)                                                SIZIN
   70 FORMAT(//17H  DATA FOR SUBARCI3 / 5X,5HSREF=E12.6,4X,9HNOZ.AREA=E1 SIZIN
     12.6,9X,4HISP=F7.3,9X,9HTH. MULT=F5.2,12X, 8HDELTA T=F9.5/         SIZIN
     210H PRT MULT=F5.1,11X,9HATM.OPTN=I2,14X,9HCNT.MODE=I3,13X,9HAER.OP SIZIN
     *TN=I3,13X,9HPRO OPTN=I3/ 5X,5HQMAX=F7.2,13X,5HGMAX=F6.3,          SIZIN
     410X,9HMAX LIFT=E12.6,                                             SIZIN
     5                                                       5X,8HHEAT  SIZIN
     6 RT=E12.6,7X,6HGMDOT=E13.6/3X,7HALFMAX=F7.3,11X,7HPHIMAX=,F7.3    SIZIN
     7,11X, 7HXCGREF=F8.3,10X, 7HZCGREF=F7.3,13X, 5HXENG=F8.3/          SIZIN
     * 5X, 5HZENG=F7.3,12X,6HXTAIL=F8.3,12X,5HDREF=F7.3,11X,6HREMAX=    SIZIN
     * E12.4,11X,6HFRATE=E12.4/                                         SIZIN
     X15H  TABLE NUMBERS/4X,6HAERO AI4,7H AERO BI4,7H AERO CI4,7H AERO D SIZIN
     AI4,7H AERO EI4,7H AERO FI4,7H AERO GI4,7H THRUSTI4,7H ISP LSI4,   SIZIN
     B7H    XCGI4,7H    ZCGI4/ 4X,6HWIND AI4,7H WIND BI4                SIZIN
     C,7H BASE DI4,7H    CNDI4)                                         SIZIN
  600 CONTINUE                                                          SIZIN
C       V     WRITE OUT  NEW  BOUNDARY CONDITION ARRAY                  SIZIN
  800 CALL  WRITMS(9,BNARR,400,21)                                      SIZIN
      RETURN                                                            SIZIN
 1000 CONTINUE                                                          AAA
      IPAYO =0                                                          AAA
      INTB =0                                                           AAA
      NSB  =0                                                           AAA
      NSARC=0                                                           AAA
C        SCAN  ARRAY TO DETERMINE IF IT IS A BRANCH PROBLEM  AND WHICH  AAA
C        BRANCH  CONTAINS  PAYOFF                                       AAA
      KI= 0                                                             AAA
      DO 1050  IARC=1,NARC                                              AAA
      IT= ITAB(IARC)                                                    AAA
      IF(IT.EQ.0) GO TO 1050                                            AAA
      DO 1045 J=1,IT                                                    AAA
      ICO = BNARR(KI+1)                                                 AAA
C                                                                       AAA
      IF(ICO.LE.10) GO TO 1045                                          AAA
      NJTARC = IARC                                                     AAA
      INTB = 2                                                          AAA
      NSB  =  ICO -10                                                   AAA
      NSARC= IARC -1                                                    AAA
      GO  TO  1051                                                      AAA
 1045 KI= KI +3                                                         AAA
 1050 CONTINUE                                                          AAA
      GO TO  1080                                                       AAA
C              CHECK FOR BRANCH THAT CONTAINS PAYOFF                    AAA
 1051 CONTINUE                                                          AAA
      KT=0                                                              AAA
      DO 1075 IARC=1,NARC                                               AAA
      KI = KTAB(IARC)                                                   AAA
      IF(KI.EQ.0) GO TO 1075                                            AAA
      DO 1070 J=1,KI                                                    AAA
      IF(IARC.NE.NARC) GO TO 1070                                       AAA
      LCD = IABS(ICOD(TARG(KT+1)))                                      AAA
      IF(LCD.NE.2) GO TO 1070                                           AAA
      LVD = IABS(IVOD(TARG(KT+1)))                                      AAA
      IPAYO = LVD                                                       AAA
      GO TO  1077                                                       AAA
 1070 KT=KT +2                                                          AAA
```

```
      1075 CONTINUE                                                    AAA
      1076 CALL  STPIT(53)                                             AAA
      1077 CONTINUE                                                    AAA
      1080 CONTINUE                                                    AAA
C           V         PHASE II  INTERFACE                              AAA
C           V-A        SCAN  INITIAL CONDITIONS                        AAA
           KI=0                                                        AAA
           DO 1300  IARC=1,NARC                                        AAA
           KK= ITAB(IARC)                                              AAA
           IF(KK.EQ.0) GO TO 1300                                      AAA
           DO 1100  J=1,KK                                             AAA
           ICD= BNARR(KI+1)                                            AAA
           IVD= BNARR(KI+2)                                            AAA
           VZ = BNARR(KI+3)                                            AAA
C           V-B    FIRST ARC                                           AAA
           IF(IARC.GT.1) GO TO 1200                                    AAA
           IF(IVD.NE.5) GO TO 1120                                     AAA
           BNARR(KI+3)= VV(1)                                          AAA
           GO TO 1190                                                  AAA
C           V-C     NEW INITIAL STATES                                 AAA
      1120 IF(ICD.NE.2.OR.IPASS.LE.2) GO TO 1190                       AAA
           GO TO(1190,1121,1121,1121,1122,1122,1122),IVD               AAA
      1121 BNARR(KI+3) = SQ(17,IVD-1)                                  AAA
           GO TO 1190                                                  AAA
      1122 BNARR(KI+3) = SQ(17,IVD-2)                                  AAA
           GO TO 1190                                                  AAA
      1200 IF(IARC.NE.2) GO TO 1210                                    AAA
           IF(IVD.NE.1.OR.ICD.NE.2) GO TO 1190                         FINI
           SQ(13,3)=2.                                                 FINI
           IF(IPASS.LE.2) GO TO 1190                                   FINI
C           V-D     OPTIMIZED  PITCH-OVER                              AAA
           BNARR(KI+3)= SQ(18,3)                                       AAA
           GO TO 1190                                                  AAA
      1210 IF(INTB.EQ.2.AND.IARC.GT.NSB) GO TO 1260                    AAA
      1211 IF(IARC.NE.IBESP +1) GO TO 1220                             AAA
           IF(IVD.NE.5) GO TO 1190                                     AAA
           IF(ICD.EQ.1) GO TO 1212                                     AAA
C            V-E    DROP WT.                                           AAA
           IF(ICD.LT.5.OR.ICD.GT.6) GO TO 1190                         AAA
           BNARR(KI+3) = VV(2)                                         AAA
           IF(ISOSP+1.EQ.IARC.AND.SQ(20,1).NE.0.) BNARR(KI+3)=VV(2)+ SQ(22,3) AAA
           GO TO 1190                                                  AAA
C            V-F     ARC  INITIAL WT.(ORBITER)                         AAA
      1212 BNARR(KI+3) =   SV(7)                                       AAA
           GO TO 1190                                                  AAA
C            V-G     SOLID DROP WT.                                    AAA
      1220 IF(SQ(20,1).EQ.0.) GO TO 1230                               AAA
           IF(ISOSP+1.NE.IARC) GOTO 1230                               AAA
           IF(IVD.NE.5) GO TO 1190                                     AAA
           IF(ICD.EQ.1) GO TO 1222                                     AAA
C          1 V-H    SOLID DROP WT.                                     AAA
           IF(ICD.LT.5.OR.ICD.GT.6) GO TO 1190                         AAA
           BNARR(KI+3) = SQ(22,3)                                      AAA
           GO TO 1190                                                  AAA
C           V-I     INITIAL WT. AFTER SOLID DROP  (NO PROVISION FOR THIS) AAA
      1222 GO TO 1190                                                  AAA
C           V-J     OPTIMAL BOOST  TIME   DURATION                     AAA
      1230 IF(IARC.NE.IBESP) GO TO1240                                 AAA
           IF(IVD.NE.1)  GO  TO 1240                                   AAA
           IF(ICD.NE.2)  GO  TO 1190                                   AAA
           IOPSTG=1                                                    UH
           IF(IPASS.LE.2) GO TO 1190                                   AAA
           BNARR(KI+3) = SQ(18,2)                                      FINI
           GO TO 1190                                                  AAA
      1240 CONTINUE                                                    AAA
C          VI  SPECIAL  BRANCH TRAJECTORY  LOGIC                       AAA
C      VI-A TEST  WHICH BRANCH IS ORBITER                              AAA
C     VI-B  TEST WHETHER INJECTION ARC                                 AAA
```

1328

```
      1260 IF( IORBI.GT.NSARC) GO TO 1280                                   AAA   1280
           IF( IARC  - NSARC) 1211,1211,1265                                AAA   1265   1211
      C  VI-C  IN  REENTRY PORTION  ( SECOND BRANCH)                        AAA
      1265 IF( IARC.NE.NJTARC+1) GO TO 1190                                 AAA          1190
      1266 IF(ICD.NE.5.OR .IVD.NE.5) GO TO 1190                             AAA          1190
           BNARR(KI+3) = SQ(19,4)                                           AAA
      C          ADD SOLID TO JETTISON WT OF REENTRY VEHICLE                AAA
           IF(ISOSP.EQ.NSB. AND.SQ(20,1).NE.0) BNARR(KI+3)= VV(2)+ SQ(22,3) AAA
           GO  TO  1190                                                     AAA          1190
      C VI-D   REENTRY IS FIRST BRANCH , ORBITER  IS SECOND BRANCH          AAA
      1280 IF( IARC - NSARC) 1285,1285, 1295                                AAA   1285   1295
      C VI-E  IN   REENTRY  BRANCH   TEST FOR  JETTISON ARC                 AAA
      1285 IF( IARC.NE.NSB+2) GO TO 1287                                    AAA   1287
           GO  TO  1266                                                     AAA   1266
      C VI-F     TEST  FOR   ENTRY INITIAL WT.  ARC                         AAA
      1287 IF(IARC.NE.NSB +1)  GO TO 1190                                   AAA          1190
           IF(IVD.NE.5) GO TO 1190                                          AAA          1190
           IF(ICD.EQ.1) GO TO  1289                                         AAA   1289
           IF(ICD.LT.5.OR.ICD.GT.6) GO TO 1190                              AAA          1190
      C   VI-G   ORBITER GROSS WT  IS  DROPPED                              AAA
           BNARR(KI+3) = SV(7)                                              AAA
           GO TO 1190                                                       AAA          1190
      C                                                                     AAA
      1289 BNARR(KI+3)= VV(2)                                               AAA
      1295 CONTINUE                                                         UG
      1190 KI=KI+3                                                          AAA
      1100 CONTINUE                                                         AAA
      1300 CONTINUE                                                         AAA
      C      VI     SCAN  TARGET  CONDITIONS                                AAA
           KT=0                                                             AAA
           DO 1400 IARC=1,NARC                                              AAA
           KL = KTAB(IARC)                                                  AAA
           IF(KL.EQ.0) GO TO  1400                                          AAA   1400
           DO 1390  J=1,KL                                                  AAA
           ICD = ICOD(TARG(KT+1))                                           AAA
           IVD = IABS(IVOD(TARG(KT+1)))                                     AAA
           LCD = IABS(ICD)                                                  AAA
           IF(IVD.NE.5) GO TO1380                                           AAA   1380
           IF(IARC.NE.IBECO) GO TO1380                                      AAA   1380
           IF(LCD.NE.1) GO TO 1380                                          AAA   1380
      C      VI-A    BOOSTER CUT-OFF  WEIGHT                                AAA
           TARG(KT+2) = QP(9)                                               AAA
      1380 KT =KT+2                                                         AAA
      1390 CONTINUE                                                         AAA
      1400 CONTINUE                                                         AAA
           GO  TO  401                                                      AAA   401
           END                                                              SIZIN
```

Phase I Sizing Module (SIZE) 8

CONTENTS

SUBRØUTINE
SIZE

SUBROUTINE SIZE SPECIFICATIONS

1. DESCRIPTION

Purpose:

Calls proper subroutine for sizing option specified, checks sizing convergence, and sets print and trajectory flags.

Comments:

This is the main program of the PADS-I sizing routine. The first pass through this routine may use the booster staging velocity or mass ratio as a primary parameter, subsequent passes use the staging and total velocities provided by the trajectory routines of PADS.

If a large step in gross weight is indicated, a special message is printed and a flag set for a second pass through the steepest descent trajectory program. This process is only executed once. If another gross step is required, the case is aborted.

The subroutine may also be terminated by exceeding a maximum number of iterations. This maximum is an input quantity.

Error Notes:

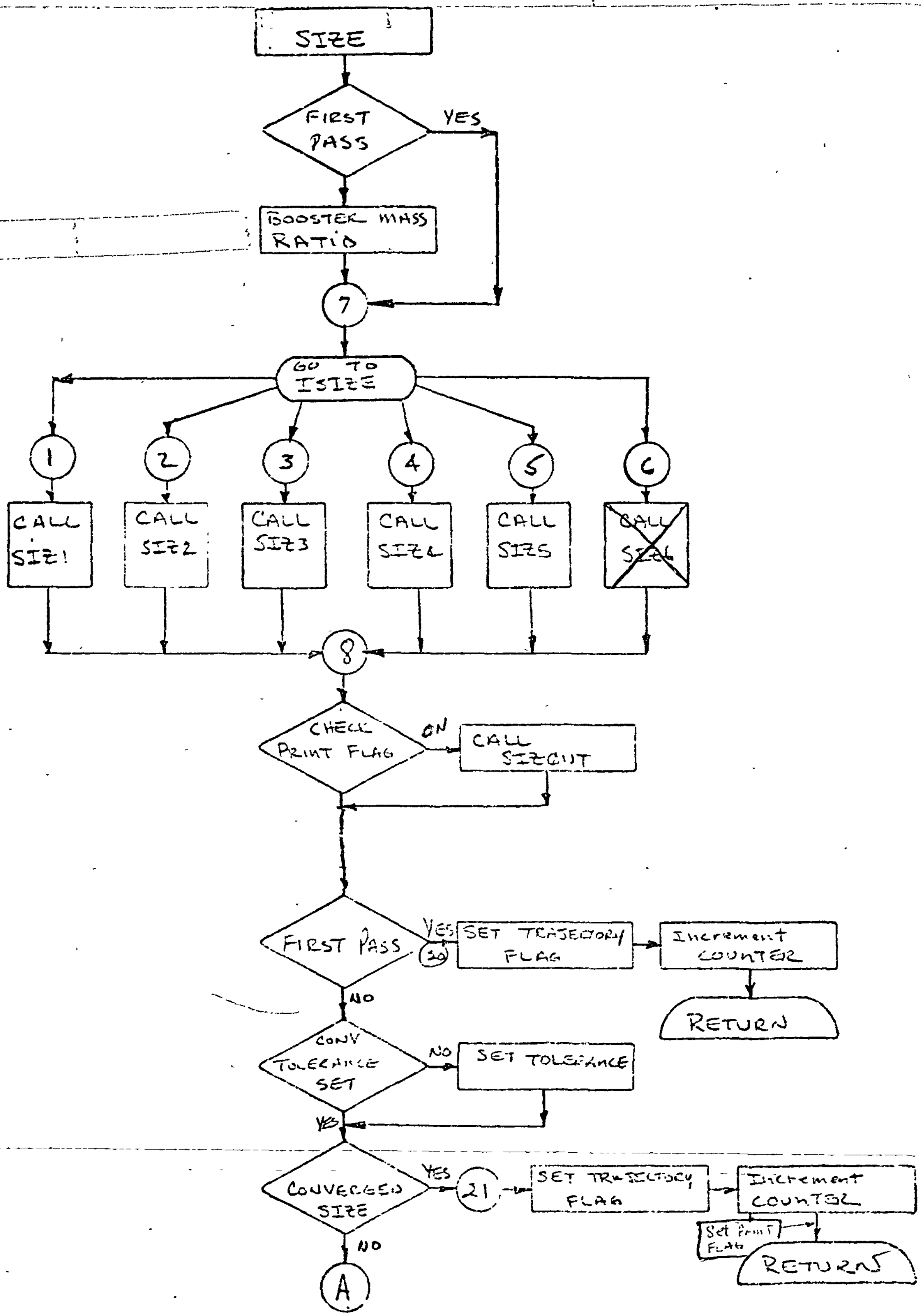
If the specific impulse of each stage is not input, a note stating this is printed and preset values of 425 and 460 sec for the booster and orbiter are used.

Iteration limit reached
This error indicates that the maximum number of iterations through the sizing routine has been exceeded. This limit may be increased through an input quantity.

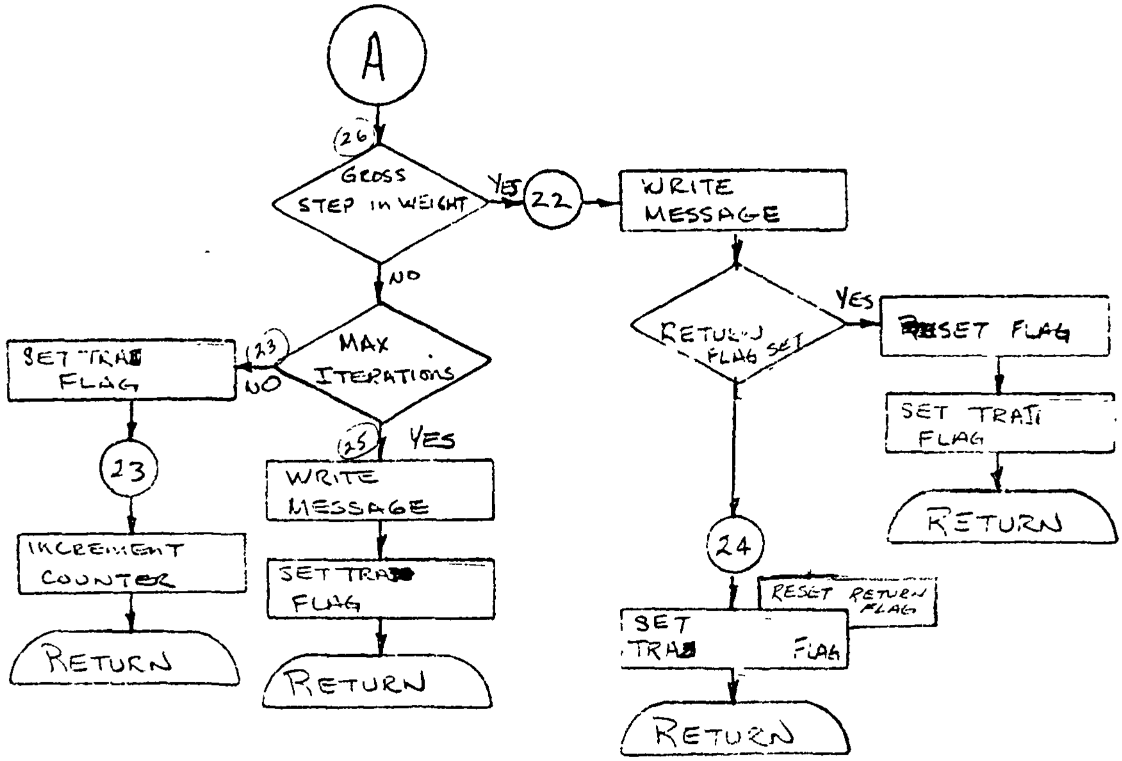
Bad starting guess on staging velocity has resulted in iteration for optimum control parameters
This error is caused by a 40% increase in the initial stage weight required. It is felt that this is an excessive amount and new control parameters would be required to converge the trajectory routines. This path may be followed once. If another gross step is indicated, the job is terminated.

If a gross weight convergence tolerance is not input, a preset value of 0.5 lb. is used.

2.0 FLOW DIAGRAMS



OVERLAY SIZE (CONT)



1334

3.0 Equations

The sizing requires an initial estimate of the booster mass ratio or staging velocity. If the staging velocity option is used, the following equation is used to determine the initial booster mass ratio

$$MUB = EXP\ (MUB/32.174/ISPB)$$

This equation is also used on the second and following passes through the sizing program since the trajectory program will be returning with the booster staging velocity (VSTG) and not the booster mass ratio.

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
BECO		I	Booster cut-off arc	/SIZING/(	314)	SIZE	I	BECO
						VEHDF	I	BECO
BSTG		I	Booster staging arc	/SIZING/(	315)	SIZE	I	BSTG
						VEHDF	I	BSTG
DAT		I	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	ACCEL	I	GR
						BL5	I	GR
						EQUA3	I	GR
						FH3	I	GR
						GEINP	I	G
						GEINP	I	GR
						GEINP	O	IG
						OUT	I	GR
						PADS1	I	GR
						PDBC	I	GR
						REU3	I	GR
						SOINP	I	GR
						SIZE	I	GR
						SIZ1	I	GR
						SIZ2	I	GR
						SIZ3	I	GR
						SIZ4	I	GR
						SOMG	I	GR
						STAU	I	GR
IPASS		M	Sizing iteration counter	/SIZING/(	291)	GEINP	O	IPASS
						PADS1	M	IPASS
						PAYO2	I	IPASS
						SIZE	M	IPASS
						SIZIN	I	IPASS
						SSSP	M	IPASS
IPSMAX		M	Maximum number of iterations	/SIZING/(	292)	SIZE	M	IPSMAX
						SSSP	I	IPSMAX
						VEHDF	I	HIPSMX
						VEHDF	O	IPSMAX
ISIZE		M	Sizing option flag 1. Fixed wlo, maximize xpl 2. Fixed xpl, minimize wlo 3. Fixed orbiter, minimize wlo 4. Fixed booster, minimize wlo 5. Fixed (t/w)l.0. Maximize xpl 6. Fixed (t/w)l.0. Determine f	/SIZING/(	283)	SIZE	M	ISIZE
						SIZOUT	I	ISIZE
ISPB		M	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		M	Orbiter vacuum specific impulse sec	/SIZING/(	296)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
ITNBW		O	Booster empty weght curve no.	/SIZING/(	317)	SIZE	O	ITNBW
						SIZ1	I	ITNBW
						SIZ2	I	ITNBW
						WTORP	I	ITNBW

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ITNOW		O	Orbiter empty weight curve no.	/SIZING/(	318)	SIZE	O	ITNOW
						SIZ1	I	ITNOW
						SIZ2	I	ITNOW
						SIZ4	I	ITNOW
						WTDRP	I	ITNOW
MUB		M	Booster mass ratio or velocity	/SIZING/(	309)	SIZE	M	MUB
						SIZ1	I	MUB
						SIZ2	I	MUB
						SIZ3	M	MUB
						SIZ4	M	MUB
MUO		M	Orbiter mass ratio	/SIZING/(	310)	SIZE	M	MUO
						SIZ1	M	MUO
						SIZ2	M	MUO
						SIZ4	M	MUO
ORBI		I	Orbiter ignition arc	/SIZING/(	316)	REU3	I	ORBI
						SIZE	I	ORBI
						VEHDF	I	ORBI
PRFLG		M	Sizing data print flag 1. Print header 2. Print identifier 3. Print data	/SIZING/(	290)	SIZE	M	PRFLG
						SIZOUT	M	PRFLG
RETFLG		W	Flag set internally to return for another pass thru the steepest descent program due to a bad guess in the sizing	/SIZE /(*	)	SIZE	W	RETFLG
SIZDAT		I	Namelist block name	/SIZE /(*SIZE)		SIZE	I	SIZDAT
SIZE		E	Main program for pads-i sizing overlay. Determines convergerce and calls proper subroutine per isize flag.	/SIZE /(*SIZE)		SIZE	E	SIZE
SIZOUT		S	Subroutine to format and print sizing data and headers	/SIZOUT/($SOUT)		SIZE	S	SIZOUT
						SIZOUT	E	SIZOUT
SIZ1		S	Sizing subroutine for fixed liftoff weight sizing option (isize=1)	/SIZ1 /($SIZ1)		SIZE	S	SIZ1
						SIZ1	E	SIZ1
						SIZ5	S	SIZ1
SIZ2		S	Sizing subroutine for fixed payload option (isize=2)	/SIZ2 /($SIZ2)		SIZE	S	SIZ2
						SIZ2	E	SIZ2
SIZ3		S	Sizing subroutine for fixed orbiter sizing option (isize=3) assumes fixed payload.	/SIZ3 /($SIZ3)		SIZE	S	SIZ3
						SIZ3	E	SIZ3
SIZ4		S	Sizing subroutine for fixed booster sizing option (isize=4) assumes fixed payload	/SIZ4 /($SIZ4)		SIZE	S	SIZ4
						SIZ4	E	SIZ4
SIZ5		S	Sizing subroutine for fixed (t/w)0 (isize=5) assumes fixed thrust	/SIZ5 /($SIZ5)		SIZE	S	SIZ5
						SIZ5	E	SIZ5
SQ		O	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
TOLWT		M	Booster liftoff weight sizing tolerance (lb)	/SIZING/(	276)	SIZE	M	TOLWT
TRAFLG		M	Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)	FRENCH	O	TRAFLG
						ITER8	O	TRAFLG
						PADS1	I	TRAFLG
						SIZE	M	TRAFLG
						SSSP	O	TRAFLG
						VEHDF	O	TRAFLG

1337

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
VSTG		I	Booster staging velocity (fps)	/SIZING/(	311)	SIZE	I	VSTG
						TRTOSZ	M	VSTG
WLO		I	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO
WLOO		M	Previous iteration value of booster liftoff weight (lb)	/SIZING/(	273)	SIZE	M	WLOO
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHOF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      PROGRAM SIZE                                                      SIZE
C                                                                       SIZE
C     PHASE I  SIZING  OVERLAY                                          SIZE
C                                                                       SIZE
      DIMENSION DAT(1)                                                  SIZE
      COMMON/GLOBAL/                                                    GLOBAL
     *GR      ,ER        ,OMGZ      ,XLAMRF    ,YMURF     ,LUM          GLOBAL
     *,JJOP(10) ,IFATAL   ,NARC      ,NBRAN     ,NFARC     ,ID(4)       GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG    ,MAXTAB                               GLOBAL
     *,GM       ,PSIRF,IPFLG1,     IPFLG2,IPFLG3,IPFLG4,INEOFL(20)      GLOBAL
     *,ITPSO  ,KSOL     ,KGLOBL(8)                                      RETAP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),     QP(14),   EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,     TLNG,                     SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOD,      DWEB,     DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,      BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,      OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,     NNB,      WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,     DVB,       MUB,      MUO,      VSTG,      WPO              SIZING
     *,JTYP    BECO       BSTG      ORBI      ITNBW    ITNOW    ,         SIZING
     * SVDPSO   SVDCON  ,IHUNT    ,IOPSTG   ,ISZO(19)                   UH
      EQUIVALENCE (DAT(1),WBO)                                          SIZE
      NAMELIST/SIZDAT/                                                  SIZE
     *WBO,   WLOD,      DWEB,      DWEO,      TOLWT,     WPB,    TWRAT2, UH
     *BK1,     BK2,       BK3,      BK4,      ISIZE,     TRAFLG,  TWRATO, SIZE
     *OK1,     OK2,       OK3,      OK4,      PRFLG,     IPASS,   IPSMAX, SIZE
     *AEXIT,   TVACO,     NO,       WFO,      IDVEL,     ISPO,    ISPB,   SIZE
     *XPL,     TVACB,     NNB,      WEO,      WEB,       WO,      WLO,    SIZE
     *DVO,     DVB,       MUB,      MUO,      VSTG,      WPO              SIZE
C                                                                       SIZE
C     FIRST  PASS TEST                                                  SIZE
C                                                                       SIZE
      IF(IPASS.EQ.1) GO TO 200                                          SIZE
      MUB = EXP(VSTG/ GR    /ISPB)                                      SIZE
      GO TO 7                                                           SIZE
C                                                                       SIZE
C     CHECK MR FOR DELTA V INPUT                                        SIZE
C                                                                       SIZE
  200 CONTINUE                                                          SIZE
      ISIZE = DAT(12)                                                   SIZE
      IPSMAX= DAT(21)                                                   SIZE
      ITNBW = DAT(46)                                                   SIZE
      ITNOW = DAT(47)                                                   SIZE
      WRITE(6,SIZDAT)                                                   UH
      SQ(1,1) = BECO                                                    SIZE
      SQ(1,2) = BSTG                                                    SIZE
      SQ(1,3) = ORBI                                                    SIZE
      IF(MUB.LE.0.) MUB=VSTG                                            UH
      IF(ISPB.NE.0..AND.MUB.GE.15.) MUB=EXP(MUB/ GR    /ISPB)           SIZE
      IF(ISPB) 202,202,201                                              SIZE
  201 CONTINUE                                                          SIZE
      IF(ISPO.NE.0..AND.MUO.GE.15.) MUO=EXP(MUO/ GR    /ISPO)           SIZE
      IF(ISPO) 204,204,7                                                SIZE
C                                                                       SIZE
C     SET DEFAULT VALUE OF ISP                                          SIZE
C                                                                       SIZE
  202 CONTINUE                                                          SIZE
      WRITE(6,203)                                                      SIZE
  203 FORMAT(45H SPECIFIC IMPULSE NOT INPUT DEFAULT VALUE USED)         SIZE
      ISPB = 425.                                                       SIZE
      GO TO 200                                                         SIZE
  204 CONTINUE                                                          SIZE
      WRITE(6,203)                                                      SIZE
      ISPO = 460.                                                       SIZE
      GO TO 201                                                         SIZE
C                                                                       SIZE
C     VEHICLE SIZING SUBROUTINES                                        SIZE
C                                                                       SIZE
    7 GO TO (1,2,3,4,5) ISIZE                                           SIZE
```

```
      1 CALL SIZ1                                                       SIZE
        GO TO 8                                                         SIZE
      2 CALL SIZ2                                                       SIZE
        GO TO 8                                                         SIZE
      3 CALL SIZ3                                                       SIZE
        GO TO 8                                                         SIZE
      4 CALL SIZ4                                                       SIZE
        GO TO 8                                                         SIZE
      5 CALL SIZ5                                                       SIZE
C                                                                       SIZE
C       VEHICLE PRINT TEST ROUTINE                                      SIZE
C                                                                       SIZE
      8 IF(PRFLG.NE.0) CALL SIZOUT                                      SIZE
        IF (IPASS.EQ.1) GO TO 20                                        SIZE
        IF(TRAFLG.GE.1.)  GO TO 999                                     SIZE
C                                                                       SIZE
C       VEHICLE WT CONVERGENCE TEST                                     SIZE
C                                                                       SIZE
        IF (TOLWT.EQ.0.0) TOLWT = 10.0                                  SIZE
        IF(ABS(WLOO - WLO) - TOLWT) 21,21,26                            SIZE
     26 IF (ABS(WLOO -WLO)/WLO .GT. 0.40)  GO TO 22                     SIZE
        IF (IPASS.LT.IPSMAX)  GO TO 23                                  SIZE
C       ITERATION LIMIT REACHED                                         SIZE
C                                                                       SIZE
     25 WRITE(6,103)                                                    SIZE
    103 FORMAT(32H MAXIMUM NO. ITERATIONS EXCEEDED)                     SIZE
        TRAFLG  = 2.                                                    SIZE
        RETURN                                                          SIZE
C                                                                       SIZE
C       NOMINAL EXIT TO O-L ROUTINE                                     SIZE
C                                                                       SIZE
     23 TRAFLG  = 0.                                                    SIZE
        IPASS = IPASS + 1                                               SIZE
        GO TO 11                                                        SIZE
C                                                                       SIZE
C       A GROSS STEP IN STAGE SIZE HAS BEEN TAKEN                       SIZE
C       RETURN FOR NEW OPTIMUM CONTROL HISTORY- ONE TIME ONLY           SIZE
C                                                                       SIZE
     22 CONTINUE                                                        SIZE
        WRITE(6,102)                                                    SIZE
    102 FORMAT(96H BAD STARTING GUESS ON STAGING VELOCITY HAS RESULTED IN SIZE
       *ITERATION FOR OPTIMUM CONTROL PARAMETERS  )                     SIZE
        IF(RETFLG.EQ.1.0) GO TO 24                                      SIZE
        RETFLG = 1.0                                                    SIZE
        TRAFLG  = 0.                                                    SIZE
        CALL TAMPAR                                                     SIZE
        RETURN                                                          SIZE
C                                                                       SIZE
C       SECOND PASS THRU STEEPEST DESCENT SOLUTION WITH NO CONVERGENCE  SIZE
C       TERMINATED FOR BAD STAGING GUESS                                SIZE
C                                                                       SIZE
     24 TRAFLG  = 2.                                                    SIZE
        RETFLG = 0.                                                     SIZE
        CALL TAMPAR                                                     SIZE
        RETURN                                                          SIZE
C                                                                       SIZE
C       CONVERGED TRAJECTORY AND SHUTTLE SIZING                         SIZE
C       SET FLAGS FOR FINAL PASS THRU O-L ROUTINE                       SIZE
C       SET FLAGS FOR FINAL VEHICLE TRAJECTORY AND PRINT                SIZE
C                                                                       SIZE
     21 TRAFLG = 2.                                                     SIZE
        WRITE(6,9) ISIZE                                                SIZE
      9 FORMAT( 23H PHASE I SIZING OPTION ,I2, 10H CONVERGED   )        SIZE
        PRFLG = 1.                                                      SIZE
        CALL TAMPAR                                                     SIZE
        GO TO 8                                                         SIZE
     20 TRAFLG  = 0.                                                    SIZE
        IPASS = IPASS + 1                                               SIZE
     11 CONTINUE                                                        SIZE
```

```
143.         WLOO = WLO                                                    SIZE
144.         CALL TAMPAR                                                   SIZE
145.     999 CONTINUE                                                      SIZE
146.         RETURN                                                        SIZE
147.         END                                                           SIZE
```


1340

SUBRØUTINE SIZ1

Subroutine SIZ1 Specifications

1.0 DESCRIPTION

Purpose:

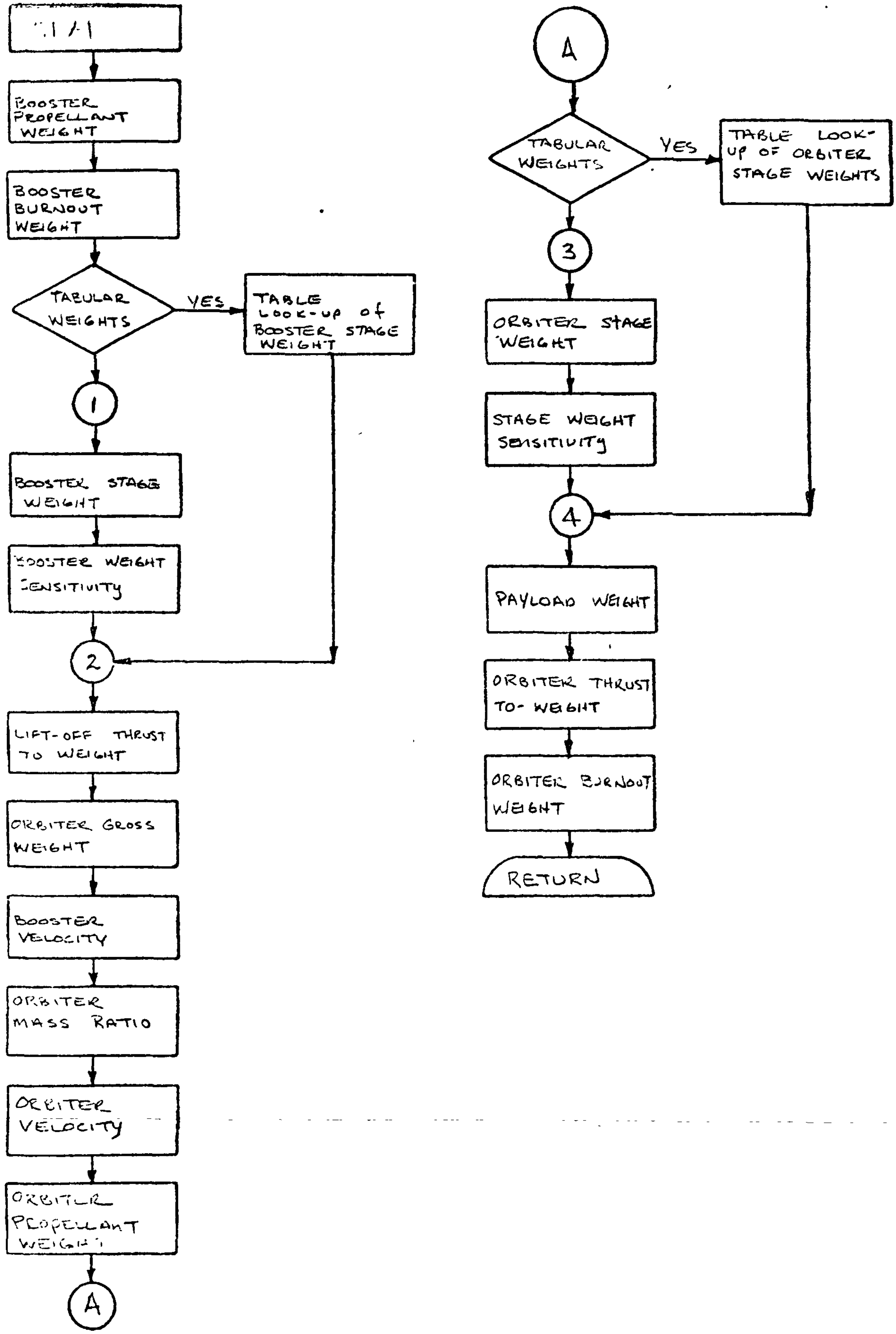
To size a two stage space shuttle with a fixed gross lift-off weight.

Comments:

This routine will size a two stage launch vehicle with a fixed initial weight to perform a given mission. The payload weight is allowed to vary. The two stages are allowed to vary and are described by input weights in a tabular format as a function of propellant weight or the stage weights may be determined by inputing coefficients to a generalized weight scaling equation.

Data transmission to and from this routine is handled by CØMMØN/SIZING/'

2.0 FLOW DIAGRAM



3.0 EQUATIONS

The fixed lift-off weight sizing equations are based on impulsive velocity relationships and are solved to maximize the payload delivery capability for the given gross initial weight.

The first quantity calculated is the booster propellant weight using the initial estimate of the booster mass ratio

$$WPB = WLO\ (MUB-1)/MUB$$

where WLO is the initial gross weight

and MUB is the estimate of the booster stage mass ratio.

The booster burnout weight is determined from

$$WBO = WLO - WPB$$

The booster stage weight is determined from the following generalized weight equation if the coefficients are input. If the coefficients are not input, the stage weight may be input in a tabular format as a function of the stage propellant weight

$$WEB = BK1 + BK2(WPB) + BK3(WPB)^{1/3} + BK4(WPB)^{2/3}$$

where BK1 through BK4 are input quantities.

The sensitivity of the booster stage weight to the booster propellant weight is determined by differentiation of the equation above to yield;

$$DWEB = BK2 + 1/3\ BK3(WPB)^{-2/3} + 2/3\ BK4(WPB)^{-1/3}$$

A similar expression is obtained from the tabular weight option if it is used.

The booster thrust to weight ratio at lift-off is calculated from

$$TWRATO = N(TVAC - AEXIT\ (2116.217))/WLO$$

and the booster inpulse velocity from

$$DVB = 32.2\ (ISPB)\ ALOG\ (MUB)$$

The initial orbiter weight is found from

$$W/O = WBO - WEB$$

The orbiter mass ratio required to provide the desired mission velocity is determined from

$$MUO = EXP((IDVEL - DVB)/(32.2\ ISPO).$$

The velocity contribution of the orbiter stage to the total mission velocity is

$$DVO = 32.2\ ISPO\ ALOG\ (MUO).$$

The orbiter propellant weight required is found from

$$WPO = WO\ (MUO - 1)/MUO.$$

The orbiter stage weight may be determined by the generalized weight equation used to determine the booster weight or may be input in a tabular format. If the generalized equation is used, the coefficients must be input.

$$WEO = OK1 + OK2(WPO) + OK3(WPO)^{1/3} + OK4(WPO)^{2/3}$$

The orbiter stage weight sensitivity is determined from

$$DWEO = OK2 + 1/3\ OK3(WPO)^{-2/3} + 2/3\ OK4(WPO)^{-1/3}$$

The payload weight that corresponds to this estimate of the total mission velocity and booster staging velocity is found from

$$XPL - WO - WPO - WEO$$

The orbiter initial vacuum thrust to weight ratio and final burnout weight determinat: conclude the SIZ1 routine

$$TWRAT2 = NO\ (TVACO)/WO$$

$$WFO = WO - WPO$$

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AEXIT		I	Booster engine exit area (ft**2)	/SIZING/(	293)	SIZ1	I	AEXIT
						SIZ2	I	AEXIT
						SIZ4	I	AEXIT
						SIZ5	I	AEXIT
						TAMPAR	I	AEXIT
BK1		I	Value of constant weight in booster stage weight equation	/SIZING/(	279)	SIZ1	I	BK1
						SIZ2	I	BK1
						WTORP	I	BK1
BK2		I	Value of linear term coefficient in booster stage weight equation	/SIZING/(	280)	SIZ1	I	BK2
						SIZ2	I	BK2
						WTORP	I	BK2
BK3		I	Value of 1/3-power term coefficient in booster stage weight equation	/SIZING/(	281)	SIZ1	I	BK3
						SIZ2	I	BK3
						WTORP	I	BK3
BK4		I	Value of 2/3-power term coefficient in booster stage weight equation	/SIZING/(	282)	SIZ1	I	BK4
						SIZ2	I	BK4
						WTORP	I	BK4
DVB		M	Booster idel velocity (fps)	/SIZING/(	308)	SIZOUT	I	DVB
						SIZ1	M	DVB
						SIZ2	M	DVB
						SIZ3	O	DVB
						SIZ4	M	DVB
DVO		O	Orbiter ideal velocity (fps)	/SIZING/(	307)	SIZOUT	I	DVO
						SIZ1	O	DVO
						SIZ2	O	DVO
						SIZ3	M	DVO
						SIZ4	O	DVO
						TRTOSZ	O	DVO
DWEB		M	Sensitivity of booster stage weight to propellant weight (lb/lb)	/SIZING/(	274)	PAYLOD	I	DWEB
						SIZOUT	I	DWEB
						SIZ1	M	DWEB
						SIZ2	M	DWEB
						STAU	I	DWEB
						WTORP	M	DWEB
DWEO		M	Sensitivity of orbiter stage weight to propellant weight (lb/lb)	/SIZING/(	275)	PAYLOD	I	DWEO
						SIZOUT	I	DWEO
						SIZ1	M	DWEO
						SIZ2	M	DWEO
						SIZ4	M	DWEO
						WTORP	M	DWEO
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	ACCEL	I	GR
						BL5	I	GR
						EQUA3	I	GR
						FH3	I	GR
						GEINP	I	G
						GEINP	I	GR
						GEINP	O	IG
						OUT	I	GR
						PADS1	I	GR
						PDBC	I	GR
						REU3	I	GR
						SDINP	I	GR
						SIZE	I	GR
						SIZ1	I	GR
						SIZ2	I	GR
						SIZ3	I	GR
						SIZ4	I	GR
						SOMG	I	GR
						STAU	I	GR
IDVEL		I	Total ideal velocity required to orbit (fps)	/SIZING/(	297)	SIZ1	I	IDVEL
						SIZ2	I	IDVEL
						SIZ3	I	IDVEL
						SIZ4	I	IDVEL
						TRTOSZ	O	IDVEL

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
ITNBW		I	Booster empty weght curve no.	/SIZING/(	317)	SIZE	O	ITNBW
						SIZ1	I	ITNBW
						SIZ2	I	ITNBW
						WTDRP	I	ITNBW
ITNOW		I	Orbiter empty weight curve no.	/SIZING/(	318)	SIZE	O	ITNOW
						SIZ1	I	ITNOW
						SIZ2	I	ITNOW
						SIZ4	I	ITNOW
						WTDRP	I	ITNOW
MUB		I	Booster mass ratio or velocity	/SIZING/(	309)	SIZE	M	MUB
						SIZ1	I	MUB
						SIZ2	I	MUB
						SIZ3	M	MUB
						SIZ4	M	MUB
MUO		M	Orbiter mass ratio	/SIZING/(	310)	SIZE	M	MUO
						SIZ1	M	MUO
						SIZ2	M	MUO
						SIZ4	M	MUO
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT	I	NNB
						SIZ1	I	NNB
						SIZ2	I	NNB
						SIZ4	I	NNB
						SIZ5	I	NNB
						TAMPAR	I	NNB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT	I	NO
						SIZ1	I	NO
						SIZ2	I	NO
						SIZ3	I	NO
						SIZ4	I	NO
						TAMPAR	I	NO
OK1		I	Same as bk1 except for orbiter	/SIZING/(	286)	SIZ1	I	OK1
						SIZ2	I	OK1
						SIZ4	I	OK1
						WTDRP	I	OK1
OK2		I	Same as bk2 except for orbiter	/SIZING/(	287)	SIZ1	I	OK2
						SIZ2	I	OK2
						SIZ4	I	OK2
						WTDRP	I	OK2
OK3		I	Same as bk3 except for orbiter	/SIZING/(	288)	SIZ1	I	OK3
						SIZ2	I	OK3
						SIZ4	I	OK3
						WTDRP	I	OK3
OK4		I	Same as bk4 except for orbiter	/SIZING/(	289)	SIZ1	I	OK4
						SIZ2	I	OK4
						SIZ4	I	OK4
						WTDRP	I	OK4
SIZ1		E	Sizing subroutine for fixed liftoff weight sizing option (isize=1)	/SIZ1	/($SIZ1)	SIZE	S	SIZ1
						SIZ1	E	SIZ1
						SIZ5	S	SIZ1

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
TWRATO		O	Liftoff thrust-to-weight ratio	/SIZING/(	285)	SIZOUT	I	TWRATO
						SIZ1	O	TWRATO
						SIZ2	O	TWRATO
						SIZ4	O	TWRATO
						SIZ5	I	TWRATO
TWRAT2		O	Second stOge thrust-to-weight ratio	/SIZING/(	278)	SIZOUT	I	TWRAT2
						SIZ1	O	TWRAT2
						SIZ2	O	TWRAT2
						SIZ3	O	TWRAT2
						SIZ4	O	TWRAT2
WBO		M	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
WEB		M	Booster stage weight (lb)	/SIZING/(	304)	PAYLOD	I	WEB
						SIZOUT	I	WEB
						SIZ1	M	WEB
						SIZ2	M	WEB
						SIZ4	I	WEB
						TAMPAR	I	WEB
						WTORP	M	WEB
WEO		M	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTORP	M	WEO
WFO		O	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WLO		I	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO
WO		M	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WPB		M	Booster propellant weight (lb)	/SIZING/(	277)	SIZOUT	I	WPB
						SIZ1	M	WPB
						SIZ2	M	WPB
						SIZ4	I	WPB
						TAMPAR	I	WPB
						WTDRP	M	WPB
WPO		M	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTDRP	M	WPO
XPL		O	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL

```
      SUBROUTINE SIZ1                                                   SIZ1
C                                                                       SIZ1
C     CONSTANT LIFT-OFF WEIGHT SIZING                                   SIZ1
C                                                                       SIZ1
      COMMON/GLOBAL/                                                    GLOBAL
     *GR        ,ER        ,DMGZ      ,XLAMRF    ,YMURF     ,LUM        GLOBAL
     *,JJOP(10) ,IFATAL    ,NARC      ,NBRAN     ,NFARC     ,ID(4)      GLOBAL
     *,KTAB(20),ITAB(20)   ,SIG       ,MAXTAB                           GLOBAL
     *,GM       ,PSIRF,IPFLG1,         IPFLG2,IPFLG3,IPFLG4,INEQFL(20)  GLOBAL
     *,ITPSO ,KSOL     ,KGLOBL(8)                                       RETAP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),      QP(14),   EROR,     PZ(5),    VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,     TLNG,                     SIZING
C        PHASE I SIZING PARAMERERS                                      SIZING
     *WBO,     WLOD,      DWEB,     DWEO,     TOLWT,    WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,      BK4,      ISIZE,    TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,      OK4,      PRFLG,    IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,      IDVEL,    ISPO,    ISPB,   SIZING
     *XPL,     TVACB,    NNB,       WEO,      WEB,      WO,      WLO,    SIZING
     *DVO,      DVB,      MUB,       MUO,      VSTG,     WPO             SIZING
     *,JTYP,    BECO ,    BSTG ,    ORBI,    ITNBW , ITNOW  ,            SIZING
     * SVDPSQ  , SVDCON  ,INUNT   ,IOPSTG  ,ISZD(19)                    UM
C                                                                       SIZ1
C     BOOSTER PROPELLANT WT                                             SIZ1
C                                                                       SIZ1
      WPB = WLO * (MUB-1.)/MUB                                          SIZ1
C                                                                       SIZ1
C     BOOSTER BURNOUT WT                                                SIZ1
C                                                                       SIZ1
      WBO = WLO - WPB                                                   SIZ1
C                                                                       SIZ1
C     BOOSTER STAGE WT AND SENSITIVITY                                  SIZ1
C                                                                       SIZ1
      IF(BK1.GT.0.0) GO TO 1                                            SIZ1
      CALL SPLIZ( ITNBW, WPB, WEB, DWEB)                                SIZ1
      GO TO 2                                                           SIZ1
    1 WEB = BK1 + BK2* WPB* BK3* WPB**0.3333* BK4* WPB**0.6667          SIZ1
      DWEB= BK2 + BK3* 0.3333* WPB**(-0.6667)+ BK4* 0.6667* WPB**       SIZ1
     *        (-0.3333)                                                 SIZ1
    2 CONTINUE                                                          SIZ1
      TWRATO =NNB*(TVACB- AEXIT * 2116.217)/WLO                         SIZ1
      DVB = GR    * ISPB * ALOG(MUB)                                    SIZ1
C                                                                       SIZ1
C     INITIAL ORBITER WT                                                SIZ1
C                                                                       SIZ1
      WO   = WBO - WEB                                                  SIZ1
C                                                                       SIZ1
C     ORBITER MASS RATIO                                                SIZ1
C                                                                       SIZ1
      MUO = EXP((IDVEL -DVB)/(GR   * ISPO))                             SIZ1
      DVO = GR    * ISPO*ALOG(MUO)                                      SIZ1
C                                                                       SIZ1
C     ORBITER PROPELLANT WT                                             SIZ1
C                                                                       SIZ1
      WPO = WO* (MUO-1.0)/MUO                                           SIZ1
C                                                                       SIZ1
C     ORBITER STAGE WT AND SENSITIVITY                                  SIZ1
C                                                                       SIZ1
      IF(OK1.GT.0.0)  GO TO 3                                           SIZ1
      CALL SPLIZ(ITNOW, WPO, WEO, DWEO)                                 SIZ1
      GO TO 4                                                           SIZ1
    3 WEO = OK1+ OK2*WPO+ OK3* WPO** 0.3333+ OK4* WPO** 0.6667          SIZ1
      DWEO= OK2+ 0.3333* OK3* WPO**(-0.6667)+ 0.6667* OK4* WPO**        SIZ1
     *        (-0.3333)                                                 SIZ1
    4 CONTINUE                                                          SIZ1
C                                                                       SIZ1
C     PAYLOAD WT                                                        SIZ1
C                                                                       SIZ1
      XPL = WO - WPO - WEO                                              SIZ1
C                                                                       SIZ1
      TWRAT2 =  NO*(TVACO)/WO                                           SIZ1
```

```
      WFO = WO - WPO                                              SIZ1
      RETURN                                                      SIZ1
      END                                                         SIZ1
```

SUBRØUTINE SIZ2

Subroutine SIZ2 Specification

1.0 DESCRIPTION

Purpose:

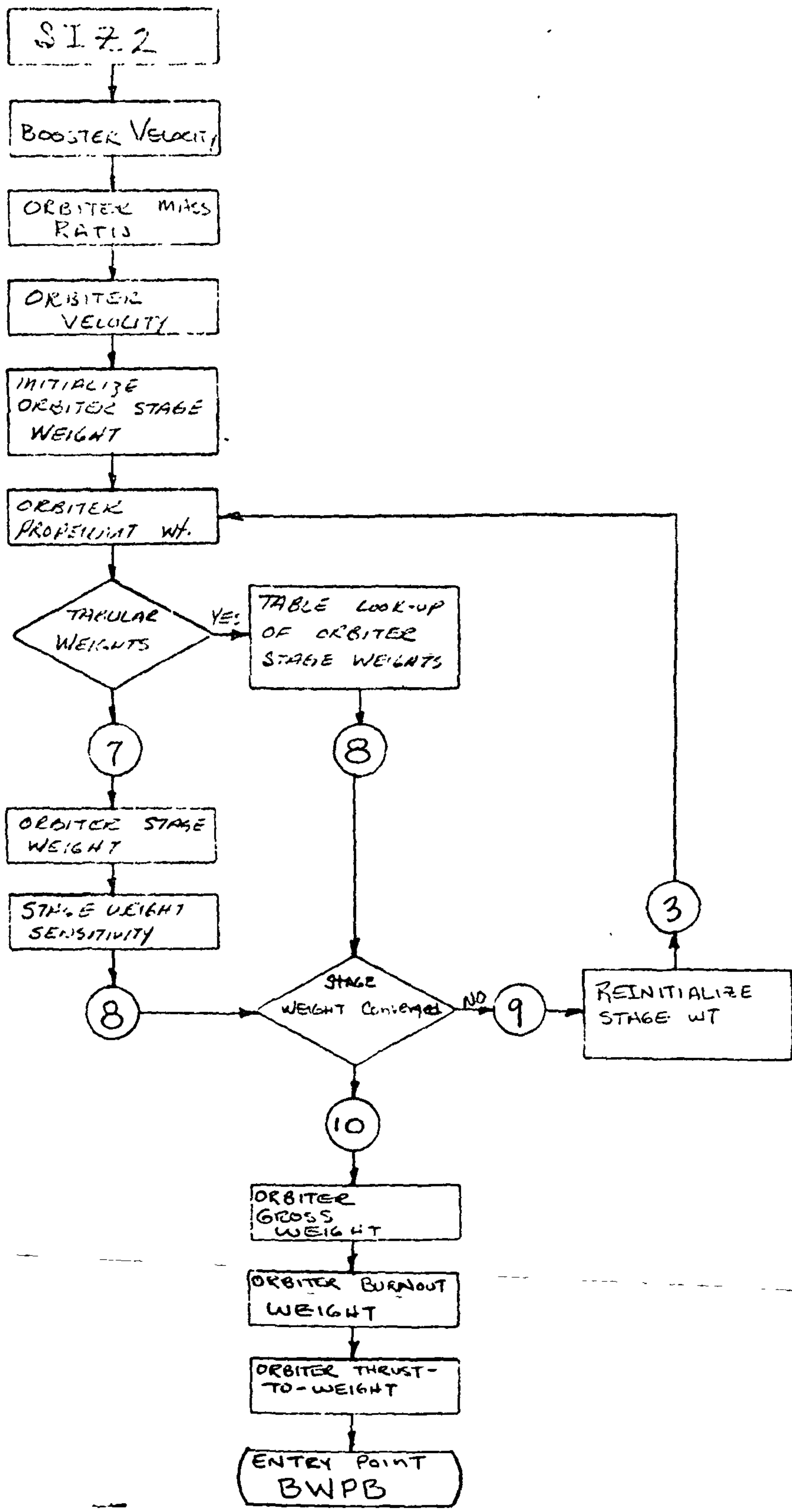
To size a two stage space shuttle with a fixed payload.

Comment:

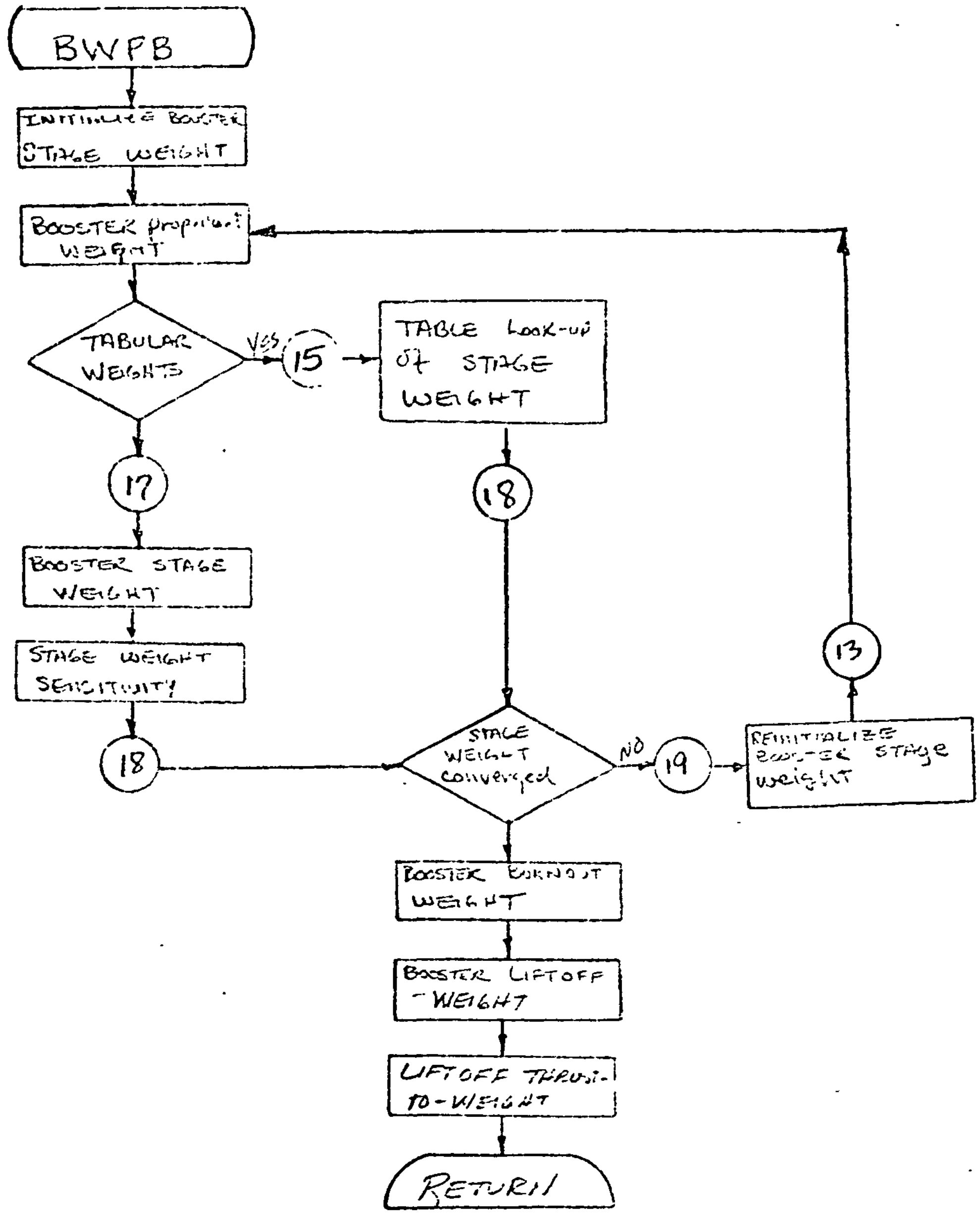
This routine interates on both stage propellant weights for the input payload weight.

Stage weights may be of a tabular input format or the coefficients of a generalized weight equation may be input.

2. FLOW DIAGRAM



SIZ2 (Cont)



BWFB
INITIALIZE BOOSTER STAGE WEIGHT
BOOSTER propellant WEIGHT
TABULAR WEIGHTS
YES
15
TABLE LOOK-UP OF STAGE WEIGHT
17
18
BOOSTER STAGE WEIGHT
STAGE WEIGHT SENSITIVITY
18
STAGE WEIGHT converged
NO
19
REINITIALIZE BOOSTER STAGE weight
13
BOOSTER BURNOUT WEIGHT
BOOSTER LIFTOFF WEIGHT
LIFTOFF THRUST-TO-WEIGHT
RETURN

3.0 EQUATIONS

The solution to the fixed payload sizing problem starts with the determination of the velocity distribution between the booster and the orbiter. The booster velocity is given by

$$DVB = 32.2 \ ISPB \ ALOG \ (MUB)$$

The required orbiter mass ratio to satisfy the mission velocity is given by

$$MUO = EXP \ ((IDNEL - DVB)/32.2/ISPO$$

AND THE ORBITER VELOCITY IS GIVEN BY

$$DVO = 32.2 \ ISPO \ ALOG \ (MUO)$$

The orbiter propellant weight is solved for iteratively from the following equations

$$WPO = (MUO - 1) \ (XPL + WEO1)/MUO$$

Where WEO1 is an initial estimate of the orbiter stage weight. An orbiter stage weight is calculated with the estimate of the orbiter propellant weight given above. Either the tabular input data of the coefficients for the general weight equation given below must be input to determine the orbiter stage weight

$$WE\emptyset = OK1 + OK2(WPO) + OK3(WPO)^{1/3} + OK4(WPO)^{2/3}$$

and stage weight sensitivity.

$$DWEO = OK2 + 1/3 \ OK3(WPO)^{-2/3} + 2/3 \ OK4(WPO)^{-1/3}$$

This value for WEO is returned to the equation above and solved until

$$WEO = WEO1$$

Once the orbiter propellant weight and stage weight has been determined, the initial orbiter gross weight may be determined from

$$WO = WEO + WPO + XPL$$

and the orbiter burnout weight from

$$WFO = WO - WPO$$

The orbiter thrust-to-weight ratio is given by

$$TWRAT2 = NO\ (TVACO)/WO$$

The booster proellant weight is solved for in a similar manner, with the booster stage weight initialized as zero

$$WEB1 = 0$$

and the propellant weight determined from

$$WPB = (MUB-1)(WO + WEB1)/MUB$$

Once again the booster stage weight corresponding to this propellant weight is solved for in a similar manner, with the booster stage weight initialized as zero

$$WEB1 = 0$$

and the propellant weight determined from

$$WPB = (MUB-1)(WO + WEB1)/MUB$$

Once again the booster stage weight corresponding to this propellant weight is determined from one of the two stage weight options available. Either the booster weight has been input in a tabular format as a function of its propellant weight or the coefficients of the generalized weight law have been input

$$WEB = BK1 + BK2(WPB) + BK3(WPB)^{1/3} + BK4(WPB)^{2/3}$$

and the sensitivity of booster weight to propellant weight

$$DWEB = BK2 + 1/3\ BK3(WPB)^{-2/3} + 2/3\ BK4(WPB)^{-1/3}$$

the iteration is continued setting WEB1 = WEB until convergence has been obtained. Once converged, the remainder of the booster parameters are determined from

$$WBO = W_o + WEB$$

$$WLO = WBO + WPB$$

$$TWRATO = N(TVAC - AEXIT(2116.217))/WLO.$$

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AEXIT		I	Booster engine exit area (ft**2)	/SIZING/(	293)	SIZ1	I	AEXIT
						SIZ2	I	AEXIT
						SIZ4	I	AEXIT
						SIZ5	I	AEXIT
						TAMPAR	I	AEXIT
BK1		I	Value of constant weight in booster stage weight equation	/SIZING/(	279)	SIZ1	I	BK1
						SIZ2	I	BK1
						WTORP	I	BK1
BK2		I	Value of linear term coefficient in booster stage weight equation	/SIZING/(	280)	SIZ1	I	BK2
						SIZ2	I	BK2
						WTORP	I	BK2
BK3		I	Value of 1/3-power term coefficient in booster stage weight equation	/SIZING/(	281)	SIZ1	I	BK3
						SIZ2	I	BK3
						WTORP	I	BK3
BK4		I	Value of 2/3-power term coefficient in booster stage weight equation	/SIZING/(	282)	SIZ1	I	BK4
						SIZ2	I	BK4
						WTORP	I	BK4
BWPB		E	Entry point for booster propellant weight loop (siz2)	/BWPB /(	$SIZ2)	SIZ2	E	BWPB
						SIZ3	S	BWPB
DVB		M	Booster idel velocity (fps)	/SIZING/(	308)	SIZOUT	I	DVB
						SIZ1	M	DVB
						SIZ2	M	DVB
						SIZ3	O	DVB
						SIZ4	M	DVB
DVO		O	Orbiter ideal velocity (fps)	/SIZING/(	307)	SIZOUT	I	DVO
						SIZ1	O	DVO
						SIZ2	O	DVO
						SIZ3	M	DVO
						SIZ4	O	DVO
						TRTOSZ	O	DVO
DWEB		M	Sensitivity of booster stage weight to propellant weight (lb/lb)	/SIZING/(	274)	PAYLOD	I	DWEB
						SIZOUT	I	DWEB
						SIZ1	M	DWEB
						SIZ2	M	DWEB
						STAU	I	DWEB
						WTORP	M	DWEB
DWEO		M	Sensitivity of orbiter stage weight to propellant weight (lb/lb)	/SIZING/(	275)	PAYLOD	I	DWEO
						SIZOUT	I	DWEO
						SIZ1	M	DWEO
						SIZ2	M	DWEO
						SIZ4	M	DWEO
						WTORP	M	DWEO
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	ACCEL	I	GR
						BL5	I	GR
						EQUA3	I	GR
						FH3	I	GR
						GEINP	I	G
						GEINP	I	GR
						GEINP	O	IG
						OUT	I	GR
						PADS1	I	GR
						PDBC	I	GR
						REU3	I	GR
						SDINP	I	GR
						SIZE	I	GR
						SIZ1	I	GR
						SIZ2	I	GR
						SIZ3	I	GR
						SIZ4	I	GR
						SOMG	I	GR
						STAU	I	GR
I		O	Do-loop indes for orbiter propellant weight determination	/SIZ2 /(	*SIZ2)	SIZ2	O	I

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
IDVEL		I	Total ideal velocity required to orbit (fps)	/SIZING/(	297)	SIZ1	I	IDVEL
						SIZ2	I	IDVEL
						SIZ3	I	IDVEL
						SIZ4	I	IDVEL
						TRTOSZ	O	IDVEL
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
ITNBW		I	Booster empty weght curve no.	/SIZING/(	317)	SIZE	O	ITNBW
						SIZ1	I	ITNBW
						SIZ2	I	ITNBW
						WTORP	I	ITNBW
ITNOW		I	Orbiter empty weight curve no.	/SIZING/(	318)	SIZE	O	ITNOW
						SIZ1	I	ITNOW
						SIZ2	I	ITNOW
						SIZ4	I	ITNOW
						WTORP	I	ITNOW
MUB		I	Booster mass ratio or velocity	/SIZING/(	309)	SIZE	M	MUB
						SIZ1	I	MUB
						SIZ2	I	MUB
						SIZ3	M	MUB
						SIZ4	M	MUB
MUO		M	Orbiter mass ratio	/SIZING/(	310)	SIZE	M	MUO
						SIZ1	M	MUO
						SIZ2	M	MUO
						SIZ4	M	MUO
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT	I	NNB
						SIZ1	I	NNB
						SIZ2	I	NNB
						SIZ4	I	NNB
						SIZ5	I	NNB
						TAMPAR	I	NNB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT	I	NO
						SIZ1	I	NO
						SIZ2	I	NO
						SIZ3	I	NO
						SIZ4	I	NO
						TAMPAR	I	NO
OK1		I	Same as bk1 except for orbiter	/SIZING/(	286)	SIZ1	I	OK1
						SIZ2	I	OK1
						SIZ4	I	OK1
						WTORP	I	OK1
OK2		I	Same as bk2 except for orbiter	/SIZING/(	287)	SIZ1	I	OK2
						SIZ2	I	OK2
						SIZ4	I	OK2
						WTORP	I	OK2
OK3		I	Same as bk3 except for orbiter	/SIZING/(	288)	SIZ1	I	OK3
						SIZ2	I	OK3
						SIZ4	I	OK3
						WTORP	I	OK3
OK4		I	Same as bk4 except for orbiter	/SIZING/(	289)	SIZ1	I	OK4
						SIZ2	I	OK4
						SIZ4	I	OK4
						WTORP	I	OK4
SIZ2		E	Sizing subroutine for fixed payload option (isize=2)	/SIZ2	/(SIZ2)	SIZE	S	SIZ2
						SIZ2	E	SIZ2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
TWRATO		O	Liftoff thrust-to-weight ratio	/SIZING/(	285)	SIZOUT	I	TWRATO
						SIZ1	O	TWRATO
						SIZ2	O	TWRATO
						SIZ4	O	TWRATO
						SIZ5	I	TWRATO
TWRAT2		O	Second st0ge thrust-to-weight ratio	/SIZING/(	278)	SIZOUT	I	TWRAT2
						SIZ1	O	TWRAT2
						SIZ2	O	TWRAT2
						SIZ3	O	TWRAT2
						SIZ4	O	TWRAT2
WBO		M	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
WEB		M	Booster stage weight (lb)	/SIZING/(	304)	PAYLOD	I	WEB
						SIZOUT	I	WEB
						SIZ1	M	WEB
						SIZ2	M	WEB
						SIZ4	I	WEB
						TAMPAR	I	WEB
						WTORP	M	WEB
WEB1		W	Iteration variable for booster stage weight	/SIZ2 /(*	)	SIZ2	W	WEB1
WEO		M	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTORP	M	WEO
WEO1		W	Iteration variable for orbiter stage weight	/SIZ2 /(*	)	SIZ2	W	WEO1
WFO		O	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WLO		M	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WO		M	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO
WPB		M	Booster propellant weight (lb)	/SIZING/(	277)	SIZOUT	I	WPB
						SIZ1	M	WPB
						SIZ2	M	WPB
						SIZ4	I	WPB
						TAMPAR	I	WPB
						WTORP	M	WPB
WPO		M	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTORP	M	WPO
X		W	Convergence error in stmt iteration	/SIZ2 /(*	)	SIZ2	W	X
XPL		I	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL

```
      SUBROUTINE SIZ2                                                   SIZ2
C                                                                       SIZ2
C     CONSTANT PAYLOAD SIZING                                           SIZ2
C                                                                       SIZ2
      COMMON/GLOBAL/                                                    GLOBAL
     *GR       ,ER         ,OMGZ      ,XLAMRF     ,YMURF     ,LUM       GLOBAL
     *,JJOP(10) ,IFATAL    ,NARC      ,NBRAN      ,NFARC     ,ID(4)     GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG    ,MAXTAB                               GLOBAL
     *,GM          ,PSIRF,IPFLG1,     IPFLG2,IPFLG3,IPFLG4,INEQFL(20)   GLOBAL
     *,ITPSO  ,KSOL      ,KGLOBL(8)                                     RETAP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,       VV(3),     OP(14),   EROR,     PZ(5),     VO,      SW(20), SIZING
     *SV(28),   SQ(37,5),  SE(11),   TLAT,     TLNG,                    SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,      WLOO,      DWEB,     DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,      BK2,       BK3,      BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,      OK2,       OK3,      OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,       WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,      TVACB,   NNB,        WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,       DVB,       MUB,      MUO,      VSTG,      WPO             SIZING
     *,JTYP,     BECO ,    BSTG ,    ORBI,     ITNBW , ITNOW  ,           SIZING
     * SVDPSQ  , SVDCON  ,IHUNT    ,IOPSTG  ,ISZD(19)                    UH
C                                                                       SIZ2
C     ORBITER MASS RATIO                                                SIZ2
C                                                                       SIZ2
      DVB = GR   * ISPB* ALOG(MUB)                                      SIZ2
      MUO = EXP((IDVEL -DVB)/(GR   * ISPO))                             SIZ2
      DVO = GR    * ISPO* ALOG(MUO)                                     SIZ2
C                                                                       SIZ2
C     ORBITER PROPELLANT WT LOOP                                        SIZ2
C                                                                       SIZ2
      WEO1=0.0                                                          SIZ2
      DO 3  I=1,15                                                      SIZ2
      WPO = (MUO -1.)*(XPL + WEO1)                                      SIZ2
      IF(OK1.GT.0.0) GO TO 7                                            SIZ2
    5 CONTINUE                                                          SIZ2
      CALL SPLIZ( ITNBW, WPB, WEB, DWEB)                                SIZ2
      GO TO 8                                                           SIZ2
    7 WEO = OK1+ OK2* WPO+ OK3*WPO**0.3333 + OK4*WPO** 0.6667           SIZ2
      DWEO= OK2+ OK3*0.3333* WPO**(-0.6667)+ OK4*0.6667* WPO**          SIZ2
     *        (-0.3333)                                                 SIZ2
    8 X = ABS(WEO1-WEO)                                                 SIZ2
      IF(X.LT.1) GO TO 10                                               SIZ2
    9 WEO1= WEO                                                         SIZ2
    3 CONTINUE                                                          SIZ2
C                                                                       SIZ2
C     ORBITER INITIAL WEIGHT                                            SIZ2
C                                                                       SIZ2
   10 CONTINUE                                                          SIZ2
      WO  = WPO + WEO + XPL                                             SIZ2
      WFO = WO - WPO                                                    SIZ2
      TWRAT2 =  NO*(TVACO)/WO                                           SIZ2
C                                                                       SIZ2
C     BOOSTER PROPELLANT WT LOOP                                        SIZ2
C                                                                       SIZ2
      ENTRY BWPB                                                        SIZ2
      WEB1= 0.0                                                         SIZ2
      DO 13 I=1,10                                                      SIZ2
      WPB = (MUB-1.0)*(WO + WEB1)                                       SIZ2
      IF(BK1.GT.0.0)  GO TO 17                                          SIZ2
   15 CONTINUE                                                          SIZ2
      CALL SPLIZ(ITNOW, WPO, WEO, DWEO)                                 SIZ2
      GO TO 18                                                          SIZ2
   17 WEB = BK1+ BK2* WPB+ BK3*WPB** 0.3337 + BK3*WPB**0.6667           SIZ2
      DWEB= BK2+ BK3*0.3333* WPB**(-0.6667) +BK4* 0.6667*WPB**          SIZ2
     *        (-0.3333)                                                 SIZ2
   18 X = ABS(WEB1-WEB)                                                 SIZ2
      IF(X.LT.1) GO TO 20                                               SIZ2
   19 WEB1= WEB                                                         SIZ2
   13 CONTINUE                                                          SIZ2
C                                                                       SIZ2
```

```
C      BOOSTER BURNOUT WT                                              SIZ2
C                                                                      SIZ2
    20 CONTINUE                                                        SIZ2
       WBO = WD +WEB                                                   SIZ2
C                                                                      SIZ2
C      BOOSTER LIFT-OFF WEIGHT                                         SIZ2
C                                                                      SIZ2
       WLO = WBO + WPB                                                 SIZ2
       TWRATO =NNB*(TVACB- AEXIT * 2116.217)/WLO                       SIZ2
       RETURN                                                          SIZ2
       END                                                             SIZ2
```

SUBRØUTINE SIZ3

Subroutine SIZ3 Specifications

1.0 DESCRIPTION

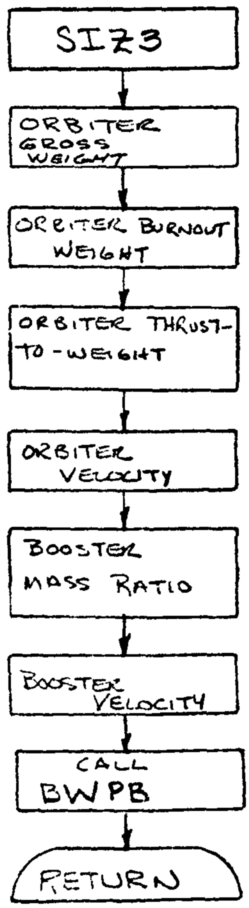
Purpose:

To size a space shuttle launch vehicle with a fixed orbiter stage and payload weight.

Comments:

This routine sizes a booster stage for a given orbiter stage size and payload weight.

The booster propellant weight is determined using the iterative loop in subroutine SIZ2 (entry point BWPB).


SIZ3
ORBITER GROSS WEIGHT
ORBITER BURNOUT WEIGHT
ORBITER THRUST-TO-WEIGHT
ORBITER VELOCITY
BOOSTER MASS RATIO
BOOSTER VELOCITY
CALL BWPB
RETURN

3.0 EQUATIONS

The orbiter is specified by its propellant weight and stage weight. The initial and final stage weights are given by

$$WO = WPO + WEO + XPL$$

$$WFO = WO - WPO$$

The thrust-to-weight ratio is given by

$$TWRAT2 = NO(TVACO)/WO$$

and the orbiter velocity is given by

$$DVO = 32.2\ ISPO\ ALOG\ (WO/WEO + XPL))$$

The required booster mass ratio is determined by

$$MUB = EXP(IDVEL - DVO)/32.2/ISPB)$$

this corresponds to a booster staging velocity of

$$DVB = 32.2\ (ISPB)\ ALOG\ (MUB).$$

With this data the propellant weight loop of subroutine SIZ2 is called (entry point BWPB) and the remaining booster parameters determined.

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
BWPB		S	Entry point for booster propellant weight loop (siz2)	/BWPB	/($SIZ2)	SIZ2	E	BWPB
						SIZ3	S	BWPB
DVB		O	Booster idel velocity (fps)	/SIZING/(	308)	SIZOUT	I	DVB
						SIZ1	M	DVB
						SIZ2	M	DVB
						SIZ3	O	DVB
						SIZ4	M	DVB
DVO		M	Orbiter ideal velocity (fps)	/SIZING/(	307)	SIZOUT	I	DVO
						SIZ1	O	DVO
						SIZ2	O	DVO
						SIZ3	M	DVO
						SIZ4	O	DVO
						TRTOSZ	O	DVO
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	ACCEL	I	GR
						BLS	I	GR
						EQUA3	I	GR
						FM3	I	GR
						GEINP	I	G
						GEINP	I	GR
						GEINP	O	IG
						OUT	I	GR
						PADS1	I	GR
						POBC	I	GR
						REU3	I	GR
						SOINP	I	GR
						SIZE	I	GR
						SIZ1	I	GR
						SIZ2	I	GR
						SIZ3	I	GR
						SIZ4	I	GR
						SOMG	I	GR
						STAU	I	GR
IDVEL		I	Total ideal velocity required to orbit (fps)	/SIZING/(	297)	SIZ1	I	IDVEL
						SIZ2	I	IDVEL
						SIZ3	I	IDVEL
						SIZ4	I	IDVEL
						TRTOSZ	O	IDVEL
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
MUB		M	Booster mass ratio or velocity	/SIZING/(	309)	SIZE	M	MUB
						SIZ1	I	MUB
						SIZ2	I	MUB
						SIZ3	M	MUB
						SIZ4	M	MUB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT	I	NO
						SIZ1	I	NO
						SIZ2	I	NO
						SIZ3	I	NO
						SIZ4	I	NO
						TAMPAR	I	NO
SIZ3		E	Sizing subroutine for fixed orbiter sizing option (isize=3) assumes fixed payload.	/SIZ3	/($SIZ3)	SIZE	S	SIZ3
						SIZ3	E	SIZ3

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
TWRAT2		O	Second st0ge thrust-to-weight ratio	/SIZING/(	278)	SIZOUT	I	TWRAT2
						SIZ1	O	TWRAT2
						SIZ2	O	TWRAT2
						SIZ3	O	TWRAT2
						SIZ4	O	TWRAT2
WEO		I	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTDRP	M	WEO
WFO		O	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WO		M	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO
WPO		I	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTDRP	M	WPO
XPL		I	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL

```
      SUBROUTINE SIZ3                                                      SIZE3
C                                                                          SIZE3
C     FIXED ORBITER SIZING                                                 SIZE3
C                                                                          SIZE3
      COMMON/GLOBAL/                                                       GLOBAL
     *GR       ,ER       ,OMGZ     ,XLAMRF    ,YMURF    ,LUM              GLOBAL
     *,JJOP(10) ,IFATAL   ,NARC     ,NBRAN    ,NFARC    ,ID(4)            GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG     ,MAXTAB                                 GLOBAL
     *,GM       ,PSIRF,IPFLG1,     IPFLG2,IPFLG3,IPFLG4,INEQFL(20)         GLOBAL
     *,ITPSO ,KSOL     ,KGLOBL(8)                                          RETAP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                              SIZING
      COMMON /SIZING/                                                      SIZING
C          PHASE II SIZING PARAMERERS                                      SIZING
     *TZ,     VV(3),    QP(14),   EROR,    PZ(5),    VQ,      SW(20),     SIZING
     *SV(28),  SQ(37,5), SE(11),  TLAT,    TLNG,                           SIZING
C         PHASE I SIZING PARAMERERS                                        SIZING
     *WBO,     WLOO,     DWEB,    DWEO,    TOLWT,    WPB,     TWRAT2,     SIZING
     *BK1,     BK2,      BK3,     BK4,     ISIZE,    TRAFLG,  TWRATO,     SIZING
     *OK1,     OK2,      OK3,     OK4,     PRFLG,    IPASS,   IPSMAX,     SIZING
     *AEXIT,   TVACO,    NO,      WFO,     IDVEL,    ISPO,    ISPB,       SIZING
     *XPL,     TVACB,   NNB,      WEO,     WEB,      WO,      WLO,        SIZING
     *DVO,      DVB,      MUB,     MUO,     VSTG,     WPO                  SIZING
     *,JTYP,    BECO,     BSTG     ORBI,   ITNBW , ITNOW  ,                SIZING
     * SVDPSQ  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZD(19)                       UH
C                                                                          SIZE3
C     INITIAL ORBITER WEIGHT                                               SIZE3
C                                                                          SIZE3
      WO = WPO + WEO + XPL                                                 SIZE3
      WFO =  WO- WPO                                                       SIZE3
      TWRAT2 =  NO*TVACO/WO                                                SIZE3
C                                                                          SIZE3
C     BOOSTER MASS RATIO                                                   SIZE3
C                                                                          SIZE3
      DVO = GR  * ISPO* ALOG(WO/(WEO +XPL))                                SIZE3
      MUB = EXP((IDVEL -DVO)/(GR  * ISPB))                                 SIZE3
      DVB = GR  *ISPB * ALOG(MUB)                                          SIZE3
C                                                                          SIZE3
C     BOOSTER PROPELLANT WT LOOP                                           SIZE3
C                                                                          SIZE3
      CALL BWPB                                                            SIZE3
      RETURN                                                               SIZE3
      END                                                                  SIZE3
```

SUBRØUTINE
SIZ4

Subroutine SIZ4 Specifications

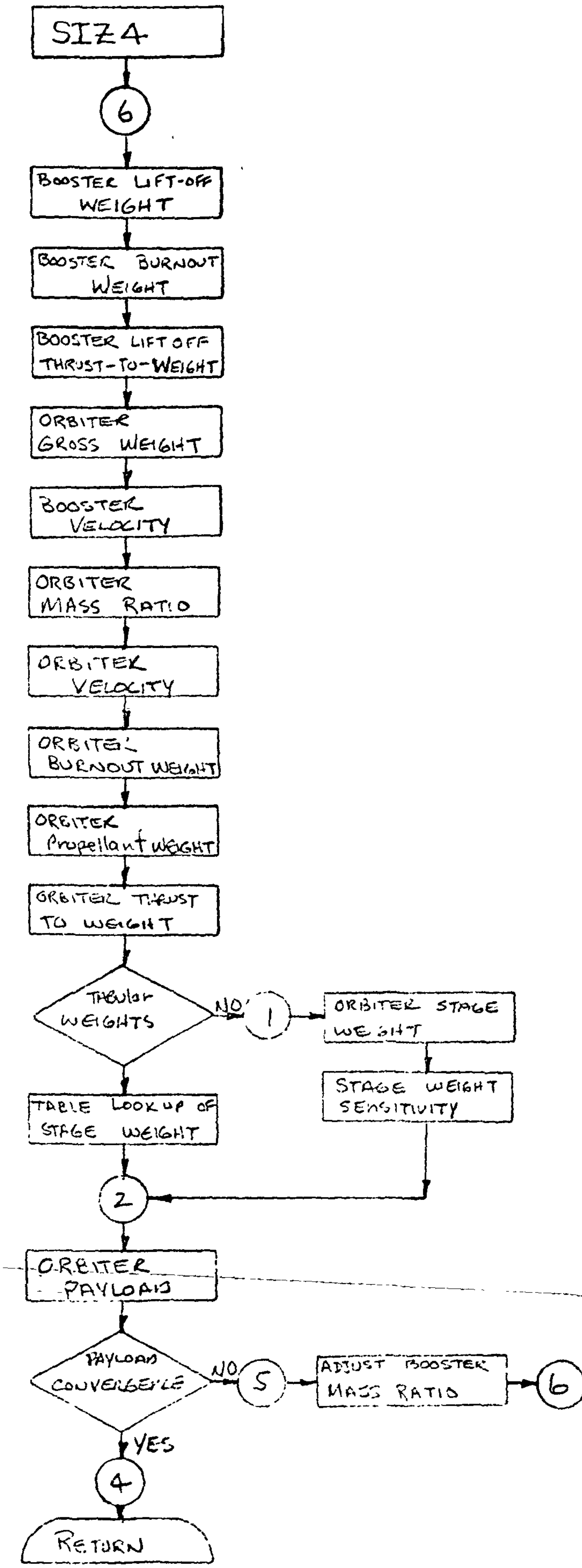
1.0 DESCRIPTION

Purpose:

To size a space shuttle using a fixed booster stage and payload weight.

Comments:

This routine starts with an initial estimate of the booster staging velocity and determines the orbiter size required. The payload value is determined and compared to the required value. If they disagree, the booster staging velocity is changed and the process repeated.


SIZ4
6
BOOSTER LIFT-OFF WEIGHT
BOOSTER BURNOUT WEIGHT
BOOSTER LIFTOFF THRUST-TO-WEIGHT
ORBITER GROSS WEIGHT
BOOSTER VELOCITY
ORBITER MASS RATIO
ORBITER VELOCITY
ORBITER BURNOUT WEIGHT
ORBITER Propellant WEIGHT
ORBITER THRUST TO WEIGHT
Tabular WEIGHTS
NO
1
ORBITER STAGE WEIGHT
STAGE WEIGHT SENSITIVITY
TABLE LOOKUP OF STAGE WEIGHT
2
ORBITER PAYLOAD
PAYLOAD CONVERGENCE
NO
5
ADJUST BOOSTER MASS RATIO
6
YES
4
RETURN

3.0 EQUATIONS

The booster lift-off weight is determined from the initial stage mass ratio estimate and the propellant weight

$$WLO = WPB(MUB)/(MUB-1).$$

Next the burnout weight and thrust-to-weight ratio are determined from

$$WBO = WLO/MUB$$

$$TWRATO = N\ (TVAC - AEXIT\ (2116.217))/WLO.$$

The corresponding booster staging velocity is given by

$$DVB = 32.2\ (ISPB)\ ALOG\ (MUB).$$

The initial orbiter weight is given by

$$WO = WBO - WEB.$$

The required mass ratio for the orbiter is given by

$$MUO = EXP\ (IDVEL - DVB)/32.2/ISPO$$

which corresponds to an orbiter velocity of

$$DVO = 32.2\ (ISPO)\ ALOG\ (MUO).$$

The orbiter vacuum thrust-to-weight ratio is given by

$$TWRATZ = NO\ (TVACO)/WO$$

and the orbiter burnout and propellant weights are given by

$$WFO = WO/MUO$$

$$WPO = WO - WFO.$$

At this point the orbiter stage weight and weight sensitivity is determined using the propellant weight and input data. If the tabular stage weight option has been

used these quantities are looked-up, whereas, if the coefficients to the generalized stage weight equation were input,

$$WEO - OK1 + OK2(WPO) + OK3(WPO)^{1/3} + OK4(WPO)^{2/3}$$

and

$$OWEO = OK2 + 1/3\ OK3(WPO)^{-2/3} + 2/3\ OK4(WPO)^{-1/3}.$$

The corresponding payload for this iteration is determined from

$$YPL = WFO - WEO$$

if this payload matches the input payload the sizing is complete. If

$$X = XPL - YPL \neq 0$$

then the booster mass ratio is changed in the proper direction by

$$MUB = MUB - X\ (MUB-1)^2/WPB$$

and the process is repeated until X is made sufficiently small.

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
AEXIT		I	Booster engine exit area (ft**2)	/SIZING/(	293)	SIZ1	I	AEXIT
						SIZ2	I	AEXIT
						SIZ4	I	AEXIT
						SIZ5	I	AEXIT
						TAMPAR	I	AEXIT
DVB		M	Booster idel velocity (fps)	/SIZING/(	308)	SIZOUT	I	DVB
						SIZ1	M	DVB
						SIZ2	M	DVB
						SIZ3	O	DVB
						SIZ4	M	DVB
DVO		O	Orbiter ideal velocity (fps)	/SIZING/(	307)	SIZOUT	I	DVO
						SIZ1	O	DVO
						SIZ2	O	DVO
						SIZ3	M	DVO
						SIZ4	O	DVO
						TRTOSZ	O	DVO
DWEO		M	Sensitivity of orbiter stage weight to propellant weight (lb/lb)	/SIZING/(	275)	PAYLOD	I	DWEO
						SIZOUT	I	DWEO
						SIZ1	M	DWEO
						SIZ2	M	DWEO
						SIZ4	M	DWEO
						WTDRP	M	DWEO
GR	g_r	I	Gravitational acceleration at surface of the earth. (FT/SEC^2)	/GLOBAL/(	1)	ACCEL	I	GR
						BL5	I	GR
						EQUA3	I	GR
						FH3	I	GR
						GEINP	I	G
						GEINP	I	GR
						GEINP	O	IG
						OUT	I	GR
						PADS1	I	GR
						PDBC	I	GR
						REU3	I	GR
						SDINP	I	GR
						SIZE	I	GR
						SIZ1	I	GR
						SIZ2	I	GR
						SIZ3	I	GR
						SIZ4	I	GR
						SOMG	I	GR
						STAU	I	GR
IDVEL		I	Total ideal velocity required to orbit (fps)	/SIZING/(	297)	SIZ1	I	IDVEL
						SIZ2	I	IDVEL
						SIZ3	I	IDVEL
						SIZ4	I	IDVEL
						TRTOSZ	O	IDVEL
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
ITNOW		I	Orbiter empty weight curve no.	/SIZING/(	318)	SIZE	O	ITNOW
						SIZ1	I	ITNOW
						SIZ2	I	ITNOW
						SIZ4	I	ITNOW
						WTDRP	I	ITNOW

1278

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
MUB		M	Booster mass ratio or velocity	/SIZING/(	309)	SIZE	M	MUB
						SIZ1	I	MUB
						SIZ2	I	MUB
						SIZ3	M	MUB
						SIZ4	M	MUB
MUO		M	Orbiter mass ratio	/SIZING/(	310)	SIZE	M	MUO
						SIZ1	M	MUO
						SIZ2	M	MUO
						SIZ4	M	MUO
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT	I	NNB
						SIZ1	I	NNB
						SIZ2	I	NNB
						SIZ4	I	NNB
						SIZ5	I	NNB
						TAMPAR	I	NNB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT	I	NO
						SIZ1	I	NO
						SIZ2	I	NO
						SIZ3	I	NO
						SIZ4	I	NO
						TAMPAR	I	NO
OK1		I	Same as bk1 except for orbiter	/SIZING/(	286)	SIZ1	I	OK1
						SIZ2	I	OK1
						SIZ4	I	OK1
						WTORP	I	OK1
OK2		I	Same as bk2 except for orbiter	/SIZING/(	287)	SIZ1	I	OK2
						SIZ2	I	OK2
						SIZ4	I	OK2
						WTORP	I	OK2
OK3		I	Same as bk3 except for orbiter	/SIZING/(	288)	SIZ1	I	OK3
						SIZ2	I	OK3
						SIZ4	I	OK3
						WTORP	I	OK3
OK4		I	Same as bk4 except for orbiter	/SIZING/(	289)	SIZ1	I	OK4
						SIZ2	I	OK4
						SIZ4	I	OK4
						WTORP	I	OK4
SIZ4		E	Sizing subroutine for fixed booster sizing option (isize=4) assumes fixed payload	/SIZ4	/($SIZ4)	SIZE	S	SIZ4
						SIZ4	E	SIZ4
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
TWRATO		O	Liftoff thrust-to-weight ratio	/SIZING/(	285)	SIZOUT	I	TWRATO
						SIZ1	O	TWRATO
						SIZ2	O	TWRATO
						SIZ4	O	TWRATO
						SIZ5	I	TWRATO
TWRAT2		O	Second st0ge thrust-to-weight ratio	/SIZING/(	278)	SIZOUT	I	TWRAT2
						SIZ1	O	TWRAT2
						SIZ2	O	TWRAT2
						SIZ3	O	TWRAT2
						SIZ4	O	TWRAT2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WBO		M	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
WEB		I	Booster stage weight (lb)	/SIZING/(	304)	PAYLOD	I	WEB
						SIZOUT	I	WEB
						SIZ1	M	WEB
						SIZ2	M	WEB
						SIZ4	I	WEB
						TAMPAR	I	WEB
						WTORP	M	WEB
WEO		M	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTORP	M	WEO
WFO		M	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WLO		M	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO
WO		M	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO
WPB		I	Booster propellant weight (lb)	/SIZING/(	277)	SIZOUT	I	WPB
						SIZ1	M	WPB
						SIZ2	M	WPB
						SIZ4	I	WPB
						TAMPAR	I	WPB
						WTORP	M	WPB
WPO		M	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTORP	M	WPO
X		W	Difference between internally determined payload and desired value (lb)	/SIZ4	/(*SIZ4)	SIZ4	W	X
XPL		I	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL
Y		W	Absolute value of delta payload	/SIZ4	/(*SIZ4)	SIZ4	W	Y

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
YPL		W	Internal value of payload used to determine stage size with one stage fixed	/SIZ4	/(*)	SIZ4	W	YPL

```
      SUBROUTINE SIZ4                                                  SIZE4
C                                                                      SIZE4
C     FIXED BOOSTER SIZING                                             SIZE4
C                                                                      SIZE4
      COMMON/GLOBAL/                                                   GLOBAL
     *GR       ,ER        ,OMGZ      ,XLAMRF     ,YMURF     ,LUM       GLOBAL
     *,JJOP(10) ,IFATAL    ,NARC      ,NBRAN      ,NFARC     ,ID(4)    GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG    ,MAXTAB                              GLOBAL
     *,GM        ,PSIRF,IPFLG1,     IPFLG2,IPFLG3,IPFLG4,INEQFL(20)    GLOBAL
     *,ITPSO ,KSOL      ,KGLOBL(8)                                     RETAP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                          SIZING
      COMMON /SIZING/                                                  SIZING
C          PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,      VV(3),     QP(14),   EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,     TLNG,                    SIZING
C         PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,     WLOO,      DWEB,     DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BKI,     BK2,       BK3,      BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OKI,     OK2,       OK3,      OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,   NNB,        WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,      MUO,      VSTG,     WPO              SIZING
     *,JTYP,    BECO,      BSTG,     ORBI,    ITNBW ,  ITNOW  ,           SIZING
     * SVOPSQ  , SVDCON  ,IHUNT    ,IOPSTG  ,ISZO(19)                   UH
    6 CONTINUE                                                         SIZE4
C                                                                      SIZE4
C     BOOSTER LIFT-OFF WT                                              SIZE4
C                                                                      SIZE4
      WLO = WPB * MUB/(MUB-1.0)                                        SIZE4
C                                                                      SIZE4
C     BOOSTER BURNOUT WT                                               SIZE4
C                                                                      SIZE4
      WBO = WLO/MUB                                                    SIZE4
      TWRATO =NNB*(TVACB- AEXIT * 2116.217)/WLO                        SIZE4
C                                                                      SIZE4
C     INITIAL ORBITER WT                                               SIZE4
C                                                                      SIZE4
      WO = WBO - WEB                                                   SIZE4
C                                                                      SIZE4
C     ORBITER MASS RATIO                                               SIZE4
C                                                                      SIZE4
      DVB = GR   * ISPB * ALOG(MUB)                                    SIZE4
      MUO = EXP((IDVEL - DVB)/(GR  * ISPO))                            SIZE4
      DVO = GR   *ISPO * ALOG(MUO)                                     SIZE4
C                                                                      SIZE4
C     ORBITER WTS                                                      SIZE4
C                                                                      SIZE4
      TWRAT2 =  NO* TVACO /WO                                          SIZE4
      WFO = WO/MUO                                                     SIZE4
      WPO = WO- WFO                                                    SIZE4
      IF(OK1.GT.0.0) GO TO 1                                           SIZE4
      CALL SPLIZ(ITNOW, WPO, WEO, DWEO)                                SIZE4
      GO TO 2                                                          SIZE4
    1 WEO = OK1+ OK2*WPO+ OK3*WPO**0.3333+ OK4*WPO**0.6667             SIZE4
      DWEO = OK2 + OK3*0.3333*WPO**(-0.6667)+ OK4*0.6667*WPO **        SIZE4
     *      (-0.3333)                                                  SIZE4
C                                                                      SIZE4
C     FIXED PAYLOAD ITERATION                                          SIZE4
C                                                                      SIZE4
    2 YPL = WFO -WEO                                                   SIZE4
      X = XPL-YPL                                                      SIZE4
      Y = ABS(X)                                                       SIZE4
      IF(Y.LT.1.) GO TO 4                                              SIZE4
      MUB = MUB -X* ((MUB - 1. )**2)/ WPB                              SIZE4
      GO TO 6                                                          SIZE4
    4 CONTINUE                                                         SIZE4
      RETURN                                                           SIZE4
      END                                                              SIZE4
```

SUBRØUTINE
SIZ5

Subroutine SIZ5 Specification

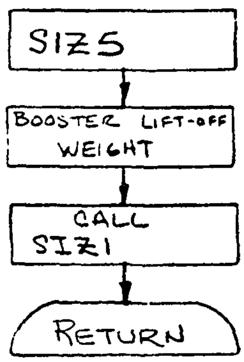
1.0 DESCRIPTION

Purpose:

To size a space shuttle with a given lift-off thrust-to-weight ratio and thrust level.

Comments:

The heart of this routine is the subroutine SIZ1.


SIZ5
BOOSTER LIFT-OFF WEIGHT
CALL SIZ1
RETURN

3.0 EQUATIONS

This routine determines the vehicle gross weight from

$$WLO = NB\ (TVACB - AEXIT\ (2116.217))/TWRATO$$

and solves the fixed gross weight problem of SIZ1.

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AEXIT		I	Booster engine exit area (ft**2)	/SIZING/(	293)	SIZ1	I	AEXIT
						SIZ2	I	AEXIT
						SIZ4	I	AEXIT
						SIZ5	I	AEXIT
						TAMPAR	I	AEXIT
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT	I	NNB
						SIZ1	I	NNB
						SIZ2	I	NNB
						SIZ4	I	NNB
						SIZ5	I	NNB
						TAMPAR	I	NNB
SIZ1		S	Sizing subroutine for fixed liftoff weight sizing option (isize=1)	/SIZ1	/($SIZ1)	SIZE	S	SIZ1
						SIZ1	E	SIZ1
						SIZ5	S	SIZ1
SIZ5		E	Sizing subroutine for fixed (t/w)0 (isize=5) assumes fixed thrust	/SIZ5	/($SIZ5)	SIZE	S	SIZ5
						SIZ5	E	SIZ5
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB
TWRATO		I	Liftoff thrust-to-weight ratio	/SIZING/(	285)	SIZOUT	I	TWRATO
						SIZ1	O	TWRATO
						SIZ2	O	TWRATO
						SIZ4	O	TWRATO
						SIZ5	I	TWRATO
WLO		O	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO

```
      SUBROUTINE SIZ5                                                    SIZ5
C                                                                        SIZ5
C     FIXED  (T/W)L.O. SIZING  - THRUST GIVEN                            SIZ5
C                                                                        SIZ5
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C          PHASE II SIZING PARAMERERS                                    SIZING
     *TZ,      VV(3),    QP(14),   EROR,     PZ(5),    VO,      SW(20),  SIZING
     *SV(28),  SO(37,5), SE(11),   TLAT,     TLNG,                       SIZING
C          PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,     DWEB,     DWEO,     TOLWT,    WPB,     TWRAT2,  SIZING
     *BK1,     BK2,      BK3,      BK4,      ISIZE,    TRAFLG,  TWRATO,  SIZING
     *OK1,     OK2,      OK3,      OK4,      PRFLG,    IPASS,   IPSMAX,  SIZING
     *AEXIT,   TVACO,    NO,       WFO,      IDVEL,    ISPO,    ISPB,    SIZING
     *XPL,     TVACB,    NNB,      WEO,      WEB,      WO,      WLO,     SIZING
     *DVO,     DVB,      MUB,      MUO,      VSTG,     WFO               SIZING
     *,JTYP,   BECO,     BSTG,     ORBI,     ITNBW,    ITNOW ,           SIZING
     * SVOPSQ  , SVOCON  ,IHUNT    ,IOPSTG   ,ISZO(19)                   UH
C                                                                        SIZ5
C     BOOSTER LIFT-OFF WEIGHT                                            SIZ5
C                                                                        SIZ5
      WLO =NNB*(TVACB- AEXIT*2116.217)/ TWRATO                           SIZ5
      CALL SIZ1                                                          SIZ5
      RETURN                                                             SIZ5
      END                                                                SIZ5
```

SUBRØUTINE SIZØUT

SUBROUTINE SIZEOUT SPECIFICATIONS

1. DESCRIPTION

Purpose:

To format and print out the PADS-I sizing data.

Comments:

This routine prints identification headers and data for all of the PADS-I sizing routines. A standard output format is used. All data is transmitted to this routine through the CØMMØN/SIZING/statement.

After the first pass through this routine, the print control flag, PRFLG, is set to print data only. The print format is shown in Table I.

TABLE I

Subroutine SIZØUT OUTPUT FORMAT

RAUS PHASE I SIZING DATA

SIZING OPTION 3 FIXED ORBITER

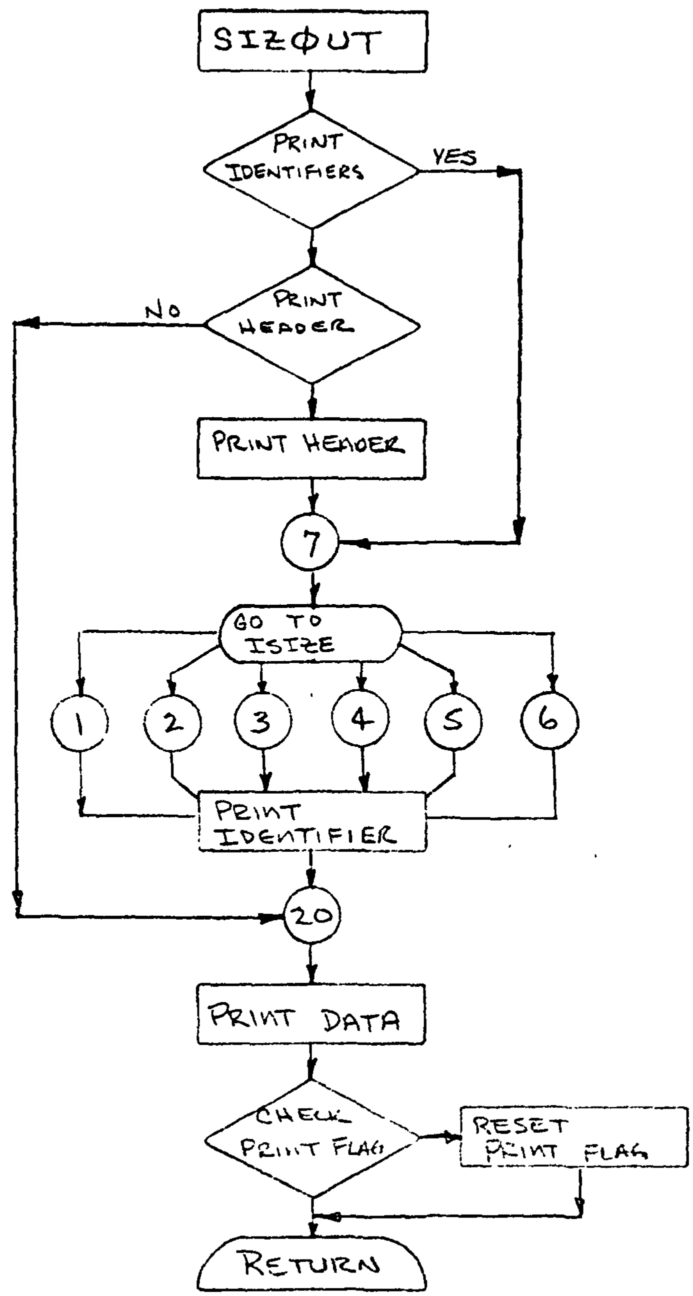
PARAMETER		BOOSTER	ORBITER
LIFT-OFF WEIGHT	(LB)	5171072	1051220
PROPELLANT WEIGHT	(LB)	3499867	827420
BURNOUT WEIGHT	(LB)	1671204	223800
STEP WEIGHT	(LB)	619984	206600
PAYLOAD WEIGHT	(LB)	1051220	17200
SPECIFIC IMPULSE	(SEC)	283	436
IMPULSIVE VELOCITY	(FPS)	10293	21718
VACUUM THRUST	(LB)	8090000	1060000
DWE/DWP		.0057	0.0000
(T/W)L.O.		1.2502	1.0084

FIRST PASS

PARAMETER		BOOSTER	ORBITER
LIFT-OFF WEIGHT	(LB)	5307158	1187158
PROPELLANT WEIGHT	(LB)	3500000	942843
BURNOUT WEIGHT	(LB)	1807158	244315
STEP WEIGHT	(LB)	620000	227116
PAYLOAD WEIGHT	(LB)	1187158	17200
SPECIFIC IMPULSE	(SEC)	283	436
IMPULSIVE VELOCITY	(FPS)	9817	22194
VACUUM THRUST	(LB)	8090000	1060000
DWE/DWP		.0057	.0288
(T/W)L.O.		1.2181	.8929

SUBSEQUENT PASSES

2. FLOW CHART


SIZØUT
PRINT IDENTIFIERS
YES
PRINT HEADER
NO
PRINT HEADER
7
GO TO ISIZE
1
2
3
4
5
6
PRINT IDENTIFIER
20
PRINT DATA
CHECK PRINT FLAG
RESET PRINT FLAG
RETURN

3.0 Equations

The only quantity calculated in this routine is the total vacuum thrust for each stage. These quantities are determined from the following equations and used for print purposes only.

Booster:

$$ABC = NB\ (TVACB)$$

Orbiter:

$$DEF = NO\ (TVACO)$$

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ABC		W	Total booster vacuum thrustused for printing sizing data	/SIZOUT/(*SIZO)		SIZOUT	W	ABC
DEF		W	Total orbiter vacuum thrustused for printing sizing data	/SIZOUT/(*SIZO)		SIZOUT	W	DEF
DVB		I	Booster idel velocity (fps)	/SIZING/(	308)	SIZOUT SIZ1 SIZ2 SIZ3 SIZ4	I M M O M	DVB DVB DVB DVB DVB
DVO		I	Orbiter ideal velocity (fps)	/SIZING/(	307)	SIZOUT SIZ1 SIZ2 SIZ3 SIZ4 TRTOSZ	I O O M O O	DVO DVO DVO DVO DVO DVO
DWEB		I	Sensitivity of booster stage weight to propellant weight (lb/lb)	/SIZING/(	274)	PAYLOD SIZOUT SIZ1 SIZ2 STAU WTORP	I I M M I M	DWEB DWEB DWEB DWEB DWEB DWEB
DWEO		I	Sensitivity of orbiter stage weight to propellant weight (lb/lb)	/SIZING/(	275)	PAYLOD SIZOUT SIZ1 SIZ2 SIZ4 WTORP	I I M M M M	DWEO DWEO DWEO DWEO DWEO DWEO
ISIZE		I	Sizing option flag 1. Fixed wlo, maximize xpl 2. Fixed xpl, minimize wlo 3. Fixed orbiter, minimize wlo 4. Fixed booster, minimize wlo 5. Fixed (t/w)1.0. Maximize xpl 6. Fixed (t/w)1.0. Determine f	/SIZING/(	283)	SIZE SIZOUT	M I	ISIZE ISIZE
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE SIZOUT SIZ1 SIZ2 SIZ3 SIZ4 TAMPAR	M I I I I I I	ISPB ISPB ISPB ISPB ISPB ISPB ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE SIZOUT SIZ1 SIZ2 SIZ3 SIZ4 TAMPAR	M I I I I I I	ISPO ISPO ISPO ISPO ISPO ISPO ISPO
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT SIZ1 SIZ2 SIZ4 SIZ5 TAMPAR	I I I I I I	NNB NNB NNB NNB NNB NNB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT SIZ1 SIZ2 SIZ3 SIZ4 TAMPAR	I I I I I I	NO NO NO NO NO NO
PRFLG		M	Sizing data print flag 1. Print header 2. Print identifier 3. Print data	/SIZING/(	290)	SIZE SIZOUT	M M	PRFLG PRFLG
SIZOUT		E	Subroutine to format and print sizing data and headers	/SIZOUT/(*SOUT)		SIZE SIZOUT	S E	SIZOUT SIZOUT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
TWRATO		I	Liftoff thrust-to-weight ratio	/SIZING/(	285)	SIZOUT	I	TWRATO
						SIZ1	O	TWRATO
						SIZ2	O	TWRATO
						SIZ4	O	TWRATO
						SIZ5	I	TWRATO
TWRAT2		I	Second st0ge thrust-to-weight ratio	/SIZING/(	278)	SIZOUT	I	TWRAT2
						SIZ1	O	TWRAT2
						SIZ2	O	TWRAT2
						SIZ3	O	TWRAT2
						SIZ4	O	TWRAT2
WBO		I	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
WEB		I	Booster stage weight (lb)	/SIZING/(	304)	PAYLOD	I	WEB
						SIZOUT	I	WEB
						SIZ1	M	WEB
						SIZ2	M	WEB
						SIZ4	I	WEB
						TAMPAR	I	WEB
						WTDRP	M	WEB
WEO		I	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTDRP	M	WEO
WFO		I	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WLO		I	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO
WO		I	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WPB		I	Booster propellant weight (lb)	/SIZING/(	277)	SIZOUT	I	WPB
						SIZ1	M	WPB
						SIZ2	M	WPB
						SIZ4	I	WPB
						TAMPAR	I	WPB
						WTDRP	M	WPB
WPO		I	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTDRP	M	WPO
XPL		I	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL
.UNO6.		O	File of all output data	/.UNO6./(S	)	BLICO	O	.UNO6.
						BNDRYC	O	.UNO6.
						CRASH	O	.UNO6.
						FRENCH	O	.UNO6.
						FXDAT	O	.UNO6.
						GEINP	O	.UNO6.
						HUNT	O	.UNO6.
						INEDIT	O	.UNO6.
						ITER8	O	.UNO6.
						MODELA	O	.UNO6.
						MOMJ	O	.UNO6.
						MPSI	O	.UNO6.
						OUT	O	.UNO6.
						PAYO2	O	.UNO6.
						PRINT	O	.UNO6.
						PRINTV	O	.UNO6.
						PRINTW	O	.UNO6.
						PRITEQ	O	.UNO6.
						PRITVA	O	.UNO6.
						PROPIN	O	.UNO6.
						PROTHR	O	.UNO6.
						PRWTSM	O	.UNO6.
						RANGE	O	.UNO6.
						S	O	.UNO6.
						SDINP	O	.UNO6.
						SIZE	O	.UNO6.
						SIZIN	O	.UNO6.
						SIZOUT	O	.UNO6.
						SOLVE	O	.UNO6.
						SPLICO	O	.UNO6.
						SPLIZ	O	.UNO6.
						SPLYNE	O	.UNO6.
						SSSP	O	.UNO6.
						STAU	O	.UNO6.
						STPIT	O	.UNO6.
						SUMOUT	O	.UNO6.
						TABIN	O	.UNO6.
						TEST	O	.UNO6.
						VEHDF	O	.UNO6.
						WTSCH	O	.UNO6.
						WTVOL	O	.UNO6.

```
      SUBROUTINE SIZOUT                                                  SIZOUT
C                                                                        SIZOUT
C     PRINT ROUTINE FOR PHASE I SIZING DATA                              SIZOUT
C                                                                        SIZOUT
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C            PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,      VV(3),      OP(14),    EROR,     PZ(5),     VO,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,      TLNG,                   SIZING
C           PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,      WLOO,      DWEB,      DWEO,      TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,       ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,       PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,       IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,      TVACB,   NNB,         WEO,       WEB,       WO,      WLO,    SIZING
     *DVO,       DVB,        MUB,       MUO,       VSTG,      WPO            SIZING
     *,JTYP      BECO      BSTG       ORBI      ITNBW  , ITNOW  ,            SIZING
     * SVOPSQ   , SVOCON   ,IHUNT     ,IOPSTG   ,ISZO(19)                    UH
C                                                                        SIZOUT
C     HEADER PRINT                                                       SIZOUT
C                                                                        SIZOUT
      IF(PRFLG.EQ.2) GO TO 7                                             SIZCUT
      IF(PRFLG.NE.1) GO TO 20                                            SIZOUT
      WRITE(6,100)                                                       SIZOUT
      PRFLG = 3.                                                         SIZOUT
    7 GO TO (1,2,3,4,5) ISIZE                                            SIZOUT
    1 WRITE(6,101)                                                       SIZOUT
      GO TO 20                                                           SIZOUT
    2 WRITE(6,102)                                                       SIZOUT
      GO TO 20                                                           SIZOUT
    3 WRITE(6,103)                                                       SIZOUT
      GO TO 20                                                           SIZOUT
    4 WRITE(6,104)                                                       SIZOUT
      GO TO 20                                                           SIZOUT
    5 WRITE(6,105)                                                       SIZOUT
   20 WRITE(6,109)                                                       SIZOUT
      ABC =NNB * TVACB                                                   SIZOUT
      DEF = NO * TVACO                                                   SIZOUT
C                                                                        SIZOUT
C     DATA PRINT                                                         SIZOUT
C                                                                        SIZOUT
      WRITE(6,110) WLO,WO,WPB,WPO,WBO,WFO,WEB,WEO,WO,XPL,                SIZOUT
     *             ISPB,ISPO,DVB,DVO,ABC,DEF,DWEB,DWEO,                  SIZOUT
     *             TWRATO, TWRAT2                                        SIZOUT
      IF(PRFLG.EQ.2) PRFLG =3.                                           SIZOUT
      WRITE(6,111)                                                       SIZOUT
  100 FORMAT(22X,24HPADS PHASE I SIZING DATA )                           SIZOUT
  101 FORMAT(20X,37HSIZING OPTION 1 FIXED LIFT-OFF WEIGHT )              SIZOUT
  102 FORMAT(20X,37HSIZING OPTION 2 FIXED PAYLOAD WEIGHT   )             SIZOUT
  103 FORMAT(20X,31HSIZING OPTION 3 FIXED ORBITER   )                    SIZOUT
  104 FORMAT(20X,31HSIZING OPTION 4 FIXED BOOSTER   )                    SIZOUT
  105 FORMAT(20X,31HSIZING OPTION 5 FIXED (T/W)L.O.)                     SIZOUT
  109 FORMAT(16X,9HPARAMETER ,19X,7HBOOSTER,11X,7HORBITER)               SIZOUT
  110 FORMAT(10X,24HLIFT-OFF WEIGHT      (LB),2(8X,F10.0),/,             SIZOUT
     *       10X,24HPROPELLANT WEIGHT    (LB),2(8X,F10.0),/,             SIZOUT
     *       10X,24HBURNOUT WEIGHT       (LB),2(8X,F10.0),/,             SIZOUT
     *       10X,24HSTEP WEIGHT          (LB),2(8X,F10.0),/,             SIZOUT
     *       10X,24HPAYLOAD WEIGHT       (LB),2(8X,F10.0),//,            SIZOUT
     *       10X,24HSPECIFIC IMPULSE    (SEC),2(8X,F10.0),/,             SIZOUT
     *       10X,24HIMPULSIVE VELOCITY  (FPS),2(8X,F10.0),/,             SIZOUT
     *       10X,24HVACUUM THRUST        (LB),2(8X,F10.0),/,             SIZOUT
     *       10X,24HDWE/DWP                  ,2(9X,F8.4) ,/,             SIZOUT
     *       10X,24H(T/W)L.O.                ,2(9X,F8.4) )               SIZOUT
  111 FORMAT(1H1)                                                        SIZOUT
      RETURN                                                             SIZOUT
      END                                                                SIZOUT
```

SUBRØUTINE TAMPAR

Subroutine TAMPAR

1.0 DESCRIPTION

Purpose:

Puts PADS-I data into format compatible with trajectory program interface.

Comments:

This routine corresponds to TAMPER in the PADS-II program (SSSP).

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
AEXIT		I	Booster engine exit area (ft**2)	/SIZING/(	293)	SIZ1	I	AEXIT
						SIZ2	I	AEXIT
						SIZ4	I	AEXIT
						SIZ5	I	AEXIT
						TAMPAR	I	AEXIT
ISPB		I	Booster vacuum specific impulse sec	/SIZING/(	299)	SIZE	M	ISPB
						SIZOUT	I	ISPB
						SIZ1	I	ISPB
						SIZ2	I	ISPB
						SIZ3	I	ISPB
						SIZ4	I	ISPB
						TAMPAR	I	ISPB
ISPO		I	Orbiter vacuum specific impulse sec	/SIZING/(	298)	SIZE	M	ISPO
						SIZOUT	I	ISPO
						SIZ1	I	ISPO
						SIZ2	I	ISPO
						SIZ3	I	ISPO
						SIZ4	I	ISPO
						TAMPAR	I	ISPO
NNB		I	Number of booster engines	/SIZING/(	302)	SIZOUT	I	NNB
						SIZ1	I	NNB
						SIZ2	I	NNB
						SIZ4	I	NNB
						SIZ5	I	NNB
						TAMPAR	I	NNB
NO		I	Number of orbiter engines	/SIZING/(	295)	SIZOUT	I	NO
						SIZ1	I	NO
						SIZ2	I	NO
						SIZ3	I	NO
						SIZ4	I	NO
						TAMPAR	I	NO
SQ		O	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		O	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TVACB		I	Booster vacuum thrust per engine lb	/SIZING/(	301)	SIZOUT	I	TVACB
						SIZ1	I	TVACB
						SIZ2	I	TVACB
						SIZ4	I	TVACB
						SIZ5	I	TVACB
						TAMPAR	I	TVACB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TVACO		I	Orbiter vacuum thrust (lb)	/SIZING/(	294)	SIZOUT	I	TVACO
						SIZ1	I	TVACO
						SIZ2	I	TVACO
						SIZ3	I	TVACO
						SIZ4	I	TVACO
						TAMPAR	I	TVACO
WBO		I	Booster burnout weight (lb)	/SIZING/(	272)	GEINP	M	SIZ
						PAYLOD	O	WBO
						SIZE	I	DAT
						SIZOUT	I	WBO
						SIZ1	M	WBO
						SIZ2	M	WBO
						SIZ4	M	WBO
						TAMPAR	I	WBO
WEB		I	Booster stage weight (lb)	/SIZING/(	304)	PAYLOD	I	WEB
						SIZOUT	I	WEB
						SIZ1	M	WEB
						SIZ2	M	WEB
						SIZ4	I	WEB
						TAMPAR	I	WEB
						WTDRP	M	WEB
WEO		I	Orbiter stage weight (lb)	/SIZING/(	303)	PAYLOD	I	WEO
						SIZOUT	I	WEO
						SIZ1	M	WEO
						SIZ2	M	WEO
						SIZ3	I	WEO
						SIZ4	M	WEO
						TAMPAR	I	WEO
						WTDRP	M	WEO
WFO		I	Orbiter burnout weight (lb)	/SIZING/(	296)	PAYLOD	M	WFO
						SIZOUT	I	WFO
						SIZ1	O	WFO
						SIZ2	O	WFO
						SIZ3	O	WFO
						SIZ4	M	WFO
						TAMPAR	I	WFO
						TRTOSZ	O	WFO
WLO		I	Booster liftoff weight (lb)	/SIZING/(	306)	PAYLOD	O	WLO
						SIZE	I	WLO
						SIZOUT	I	WLO
						SIZ1	I	WLO
						SIZ2	M	WLO
						SIZ4	M	WLO
						SIZ5	O	WLO
						TAMPAR	I	WLO
WO		I	Initial orbiter weight (lb)	/SIZING/(	305)	PAYLOD	O	WO
						SIZOUT	I	WO
						SIZ1	M	WO
						SIZ2	M	WO
						SIZ3	M	WO
						SIZ4	M	WO
						TAMPAR	I	WO
WPB		I	Booster propellant weight (lb)	/SIZING/(	277)	SIZOUT	I	WPB
						SIZ1	M	WPB
						SIZ2	M	WPB
						SIZ4	I	WPB
						TAMPAR	I	WPB
						WTDRP	M	WPB
WPO		I	Orbiter propellant weight (lb)	/SIZING/(	312)	SIZOUT	I	WPO
						SIZ1	M	WPO
						SIZ2	M	WPO
						SIZ3	I	WPO
						SIZ4	M	WPO
						TAMPAR	I	WPO
						WTDRP	M	WPO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
XPL		1	Payload weight (lb)	/SIZING/(	300)	SIZOUT	I	XPL
						SIZ1	O	XPL
						SIZ2	I	XPL
						SIZ3	I	XPL
						SIZ4	I	XPL
						TAMPAR	I	XPL

```
      SUBROUTINE TAMPAR                                                    TAMPAR
C                                                                          TAMPAR
C          SUBROUTINE TO INTERFACE WITH TRAJECTORY PROGRAM                 TAMPAR
      REAL MUB, MUO, ISPB, ISPO, IOVEL,NNB,NO                              SIZING
      COMMON /SIZING/                                                      SIZING
C           PHASE II SIZING PARAMERERS                                     SIZING
     *TZ,      VV(3),     QP(14),   EROR,    PZ(5),     V0,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,    TLNG,                       SIZING
C          PHASE I SIZING PARAMERERS                                      SIZING
     *WBO,     WLOO,      DWEB,     DWEO,    TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,      BK4,     ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,      OK4,     PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,     IOVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,   NNB,        WEO,     WEB,       WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,      MUO,     VSTG,      WPO              SIZING
     *,JTYP,    BECO ,    BSTG      ORBI,   ITNBW , ITNOW  ,              SIZING
     * SVOPSQ  , SVOCON  ,IHUNT    ,IOPSTG  ,ISZO(19)                     UH
C                                                                          TAMPAR
      VV(1)    = WLO                                                       TAMPAR
      VV(2)=WEB                                                            UH
      VV(3)=WPB                                                            UH
      QP(1)    = TVACB*NNB                                                 TAMPAR
      QP(2)    = TVACO   *NO                                               TAMPAR
      QP(3)    = ISPB                                                      TAMPAR
      QP(4)    = ISPO                                                      TAMPAR
      QP(9)    = WBO                                                       TAMPAR
      QP(10)   = XPL                                                       TAMPAR
      QP(11)   = AEXIT                                                     TAMPAR
      SV(4)    = WFO                                                       TAMPAR
      SV(5)    = WPO                                                       TAMPAR
      SV(7)    = WO                                                        TAMPAR
      SQ(4,1)=WEO                                                          OS
      RETURN                                                               TAMPAR
      END                                                                  TAMPAR
```

20 OCT 72 6.01-46

Space Shuttle Synthesis Module (SSSP) 9

CONTENTS

Subroutine	TBL2D
Subroutine	THRUP
Subroutine	THRUST
Subroutine	VEHDF
Subroutine	WTSCH
Subroutine	WTVØL

SUBRØUTINE
SSSP

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
A		D	Booster sub-sonic l/d	/DATA2X/(	1)	SSSP	D	A
						VEHDF	M	A
						VEHDF	I	ALD
IHUNT		I	Number of iterations for parameter hunt	/SIZING/(	321)	GEINP	O	IHUNT
						OPWELL	I	ELIM
						SSSP	I	IHUNT
IPASS		M	Sizing iteration counter	/SIZING/(	291)	GEINP	O	IPASS
						PADS1	M	IPASS
						PAYO2	I	IPASS
						SIZE	M	IPASS
						SIZIN	I	IPASS
						SSSP	M	IPASS
IPSMAX		I	Maximum number of iterations	/SIZING/(	292)	SIZE	M	IPSMAX
						SSSP	I	IPSMAX
						VEHDF	I	HIPSMX
						VEHDF	O	IPSMAX
SSSP		E	Main program of phase ii paos sizing program overlay(5,0)	/SSSP /($	)	SSSP	E	SSSP
SUMOUT		S	Subroutine to print summary data and calculate thrust for output purposes only	/SUMOUT/($	)	SSSP	S	SUMOUT
						SUMOUT	E	SUMOUT
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB27		I	Stored booster value of isp(i)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		M	Stored booster value of mr(i)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TO27		I	Stored orbiter value of isp(i)	/ORBINX/(	41)	SIZEMR	I	TO27
						SSSP	I	TO27
						STORE	M	TO27
						SUMOUT	I	TO27
						VEHDF	M	TO27
						WTVOL	I	TO27
TO34		O	Stored orbiter value of mr(i)	/ORBINX/(	53)	ITER8	M	TO34
						SSSP	O	TO34
						STORE	M	TO34
						TAMPER	I	TO34
						VEHDF	O	TO34
						WTVOL	M	TO34
TRAFLG		O	Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)	FRENCH	O	TRAFLG
						ITER8	O	TRAFLG
						PADS1	I	TRAFLG
						SIZE	M	TRAFLG
						SSSP	O	TRAFLG
						VEHDF	O	TRAFLG

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WGROSO		I	Orbiter gross weight	/TAMP /(	1)	SSSP	I	WGROSO
						SUMOUT	I	WGROSO
						WTVOL	M	WGROSO
WPAYLO		I	Payload weight	/SUMVW /(	16)	SSSP	I	WPAYLO
						SUMOUT	I	WPAYLO
						TAMPER	M	WPAYLO
.UN06.		O	File of all output data	/.UN06./(	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      PROGRAM SSSP                                                         SSSP
C                                                                          SSSP
C           DRIVER SUBROUTINE FOR PHASE 2 DADS SIZING PROGRAM              SSSP
C           CALCULATES FLYBACK FUEL REQUIRED                               SSSP
C                                                                          SSSP
      DIMENSION SSPDAT(1)                                                  UH
      EQUIVALENCE (SSPDAT(1),TB1)                                          CKOUT
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                              SIZING
      COMMON /SIZING/                                                      SIZING
C            PHASE II SIZING PARAMERERS                                    SIZING
     *TZ,      VV(3),     QP(14),   EROR,    PZ(5),     VQ,      SW(20),  SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,    TLNG,                         SIZING
C           PHASE I SIZING PARAMERERS                                      SIZING
     *WBO,      WLOD,      DWEB,     DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,       BK3,      BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,       OK3,      OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,       WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,   NNB,        WEO,       WEB,       WO,       WLO,    SIZING
     *DVO,       DVB,       MUB,      MUO,       VSTG,      WPO             SIZING
     *,JTYP,     BECO,      BSTG,     ORBI,    ITNBW  , ITNOW  ,            SIZING
     * SVOPSQ  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZO(19)                        UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),          ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                      CKOUT
      COMMON/ORBINY/                                                       ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,        ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27,   ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40,   ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53,   ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                  UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,        UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,         UH
     8TB83,TB84                                                            UH
      COMMON/TAMP/WGROSO,OTTOT,WFUOXO,TBTO,BTTOT                           SSSP
      REAL LBODYO,LBODYB                                                   SUMVW
      COMMON/SUMVW/WOP,            WABFUB,WFUOXB,WDRYB,WGROSB,WOTHB,       SUMVW
     1 VFUTKB,VOXTKB,VOTHB,VBODYB,VABFUB,                                  SUMVW
     2 LBODYB,SBODYB,SPLANB,WOVRSB,                                        SUMVW
     1                 WPAYLO,WDRYO,WOTHO,  WABFUD,                        SUMVW
     2 VFUTKO,VOXTKO,VCARO,VOTHO,VBODYO,                                   SUMVW
     3 LBODYO,SBODYO,SPLANO,WOVRSO                                         SUMVW
     4 ,WORBTO,WORBTB,WRTRNO,WRTRNB,WENTRO,WENTRB,WLANDB,WLANDO            SUMVW
     5 , WCONTO,WCONTB                                                     SUMVW
      REAL KIN                                                             CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                   CINPUT
      REAL NENGS                                                           CINPUT
      COMMON/CINPUT/                                                       CINPUT
     1ANENGS   ,ANTANK   ,ASRATO   ,ASWEEP   ,C(300)   ,CBBODY   ,CFUEL(6),  CINPUT
     2CHBODY   ,CLBODY   ,CSBODY   ,CSFAIR   ,CSFUTK   ,CSHORZ   ,CSOXTK   ,  CINPUT
     3CSPLAN   ,CSVERT   ,CSWING   ,CTHRST   ,CTHST2   ,DEF(5)   ,FXWOVS   ,  CINPUT
     4ISP(6)   ,ITPS     ,K(30)    ,KIN      ,LF       ,MR(6)    ,NCREW    ,  CINPUT
     5NENGS    ,NLISTO   ,NPASS    ,NWL      ,PCHAM    ,Q        ,RHOFU    ,  CINPUT
     6RHOFU2   ,RHOX     ,RHOX2    ,SBODY    ,TOL      ,TOVERC   ,TPRATO   ,  CINPUT
     7TYTAIL   ,VBODY    ,WGROSS                                              CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN    ,CTIP     ,GAL      ,GSPAN    ,  VOLCAL
     2HBODY    ,LBODY    ,RTOD     ,SFAIR    ,SFUTK    ,SHORZ    ,SOXTK    ,  VOLCAL
     3SPLAN    ,STPS(1)  ,SVERT    ,SWING    ,SXPOS    ,TDEL     ,TROOT    ,  VOLCAL
     4TTOT     ,TTOT2    ,TTOTAL   ,VBODYA   ,VBODY1   ,VBODY2   ,VCARGO   ,  VOLCAL
     5VCREW    ,VFUTK    ,VFUTK2   ,VINSTK   ,VLGBAY   ,VOTHER   ,VOXTK    ,  VOLCAL
     6VOXTK2   ,VPROP    ,VSTRUC                                              VOLCAL
      COMMON/WTCALC/     ABFSYS   ,WABFTK   ,WABFU    ,WABPR    ,WACRES   ,  WTCALC
     1WACS     ,WACSFO   ,WACSTK   ,WAERO    ,WAUXT    ,WBASIC   ,WBODY    ,  WTCALC
     2WBPUMP   ,WCARGO   ,WCOMM    ,WCONT    ,WCOVER   ,WDECAY   ,WDIST1   ,  WTCALC
     3WDIST2   ,WDOCK    ,WDPLOY   ,WDRANS   ,WDRY     ,WELCAD   ,WEMPTY   ,  WTCALC
     4WENGMT   ,WENGS    ,WENGS2   ,WFAIR    ,WFCONT   ,WFDCAY   ,WFROST   ,  WTCALC
     5WFU2(3)  ,WFUEL(6),WFUL     ,WFULOS   ,WFUNCT   ,WFUOX    ,WFURES   ,  WTCALC
     6WFUSYS   ,WFUTK    ,WFUTK2   ,WFUTOT   ,WFUTRP   ,WGASPR   ,WGNAV    ,  WTCALC
     7WHORZ    ,WHYCAD   ,WINFUT   ,WINOXT   ,WINSTK   ,WINST    ,WINSUL   ,  WTCALC
     8WJET(6)  ,WLANCH   ,WLG      ,WLOSS    ,WLRD     ,WNACEL   ,WODCAY   ,  WTCALC
     9WOIL     ,WOILRS   ,WORSUL   ,WOVERS   ,WOX(6)   ,WOX2(3)  ,WOXID    ,  WTCALC
     1WOXLOS   ,WOXRES   ,WOXSYS   ,WOXTK    ,WOXTK2   ,WOXTOT   ,WOXTRP   ,  WTCALC
     2WP       ,WPASS    ,WPAYL    ,WPERS    ,WPOWCO   ,WPOWER   ,WPOWFO   ,  WTCALC
     3WPOWRS   ,WPOWTK   ,WPPROV   ,WPREIG   ,WPROP    ,WPRSYS   ,WREFUL   ,  WTCALC
     4WRESID   ,WRESRV   ,WSEAL    ,WSECST   ,WSORCE   ,WSRTRP   ,WSTAB    ,  WTCALC
     5WSURF    ,WTABC    ,WTHRST   ,WTO      ,WTPS     ,WVERT    ,WWAIT(10),  WTCALC
     6WWET     ,WWING    ,WZROFU   ,WABTRP   ,WABRES   ,WANOTP   ,WANFTP   ,  CKOUT
```

```
      7WMNORS   ,WMNFRS   ,WACOTP   ,WACFTP   ,WPWOTP   ,WPWFTP   ,WGAS     ,     CKOUT
      8WABFUC   ,WACORS   ,WACFRS   ,WPWORS   ,WPWFRS                           CKOUT
       REAL NXFOB                                                              DATA2X
       REAL      IVACO,      IVACB,      ISLO,       ISLB,       IDVELO          DATA2X
       COMMON/DATA2X/                                                          DATA2X
      *ALD, FBPAR, IDVELO, ISLB, ISLO, IVACB, IVACO, PERISP, QMXX,             DATA2X
      1 QMXS,SFC,SLVOUT,COPIES,SYNIT,TFCTAB,TFCTRO,TOLMU,TOLTW,                DATA2X
      2 TRATIO,TWLO,TWLOI,WTOUT,FIRE,BOOTW,VCRUSE                              DATA2X
      3 ,NXFOB,PANTX,FSEC,CLVG,DRNG                                            DATA2X
      4 ,SOLID,AS,BS,SISP,SINERT,SAE,TSBO,FLYBCK                               DATA2X
      5 ,WPOREQ,WOREQ,GWREQ                                                    DATA2X
      6 ,FBFUEL,CA,CB,WFLYX,RT,R1,R3,SFC1,SFC2,SFC3,ALD1,ALD2,ALD3,VFLY1,      DATA2X
      7 VFLY2,VFLY3                                                            DATA2X
      *,TWOX(2),PNDX(10),BLOW(10),BUPP(10),STEP(11),PAYX                       POW
       EQUIVALENCE (RVAR,TWOX(2))                                              POW
       DIMENSION SKO(30),SCO(300),T04(6),T020(10),T027(6),T034(6),             ORBINX
      1 T048(10),T049(10),T050(10),T057(6),OWSAVE(10)                          CKOUT
       COMMON/ORBINX/                                                          ORBINX
      1 T01,T02,T03,T04,T05,T06,T07,T08,T09,T010,T011,T012,T013,T014,          ORBINX
      2 T015,T016,T017,T018,T019,T020,T021,T022,T023,T024,T025,T026,T027,      ORBINX
      3 T028,T029,T030,T031,T032,T033,T034,T035,T036,T037,T038,T039,T040,      ORBINX
      4 T041,T042,T043,T044,T045,T046,T047,T048,T049,T050,T051,T052,T053,      ORBINX
      5 T054, T055,T056,T057,T066,SKO,SCO,OWSAVE                               CKOUT
      6,T059,T060,T061,T062,T063,T064,T065,T067,T068,T069,T070,T071,           UH
      7 T072,T073,T074,T075,T076,T077,T078,T079,T080,T081,T082,T083,           UH
      8T084                                                                    UH
       COMMON /TRUST/                                                          SSSP
      *FVACO    ,FSLO      ,FVACLO    ,FVACS ,                                 SSSP
      *FVACB    ,FSLB      ,FSLLO     ,FSLS                                    SSSP
       COMMON/JUMPY/  JUMP, WBIG, WBOO                                         CKOUT
        COMMON/PRESET/ PRESET(57)                                              CKOUT
       DIMENSION A(57)                                                         CKOUT
       EQUIVALENCE (ALD,A(1))                                                  CKOUT
       DATA( A(I), I= 1,57)    /57*-0./                                        CKOUT
       DATA (PRESET(I),I=1,57)                                                 CKOUT
      */ 6.,      250.,      30000.,    390.,      390.,      450.,      450.,  CKOUT
      *0.81,      550.,      900.,      0.2,       0.,        6.,        5.,    CKOUT
      *1.,        1.,        0.0005,    0.001,     1.0,       1.371,     -1.,   CKOUT
      *0.,        2.,        0.,        300.,      0.,        1.,        0.,    CKOUT
      * 1.,0.,0.,10*0.,1.,9*0.,6*1./                                           UH
       COMMON/POWEL/ JJJJ(30) ,CON, JIJJ, EN,PMP(123)                          OPTSTG
       COMMON/OPWEL/ JJJJJ(30),DON, JI, DN, PNP(123)                           OPTSTG
       INTEGER DON,DN                                                          OPTSTG
       INTEGER CON,EN                                                          UH
       F(X,Y)  = EXP(Y/(32.174*X))                                             POW
       G(X,Y)  = 32.174* X * ALOG(Y)                                           POW
C                                                                              SSSP
C           FIRST PASS TEST                                                    SSSP
C                                                                              SSSP
       IF( IPASS.EQ.1 )  GO TO 1                                               POW
C                                                                              SSSP
C           ITERATION PASS                                                     SSSP
C                                                                              SSSP
       IF(IPASS.GT.IPSMAX) GO TO 13                                            UH
       CALL READMS(3,SSPDAT,1699,1)                                            POW
       IF(SW(3).GT.SW(4)) GO TO 10                                             UH
     2 CALL OVERLAY(4HSSSP,7,4,6HRECALL)                                       CKOUT
    18 IF( IOPSTG.GT.0 )  GO TO 15                                             POW
       CALL OVERLAY(4HSSSP,7,2,6HRECALL)                                       FINI
C                                                                              SSSP
C           CHECK CONVERGENCE OF SYNTHESIS LOOP                                SSSP
C                                                                              SSSP
    17 CONTINUE                                                                POW
       IF( SW(2).LE.0.5) GO TO 3                                               SSSP
C                                                                              SSSP
C           CONVERGED VEHICLE CALL FOR ORBITER WEIGHTS                         SSSP
C                                                                              SSSP
     4 CALL OVERLAY(4HSSSP,7,2,6HRECALL)                                       CKOUT
       CALL SUMOUT                                                             CKOUT
C                                                                              SSSP
C           RESET FLAGS FOR NEXT CASE                                          SSSP
C                                                                              SSSP
       IPASS = 1                                                               SSSP
       SW(2) = 0.5                                                             SSSP
```

```
      IF(INUNT.LE.0) GO TO 999                                   UH
      CALL OVERLAY(4HHUNT,7,3,0)                                 UH
      IF(CON.NE.-1) GO TO 999                                    UH
      TRAFLG=0                                                   UH
      IPASS=2                                                    UH
       SW(3)=0.                                                  UH
      GO TO 2                                                    UH
C                                                                SSSP
C          FIRST PASS LOGIC - INITIATE FLAGS                     SSSP
C                                                                SSSP
    1 SW(3)    = 0.                                              SSSP
      TRAFLG=0.                                                  UH
C                                                                SSSP
      EW=0                                                       UH
C          CALL FOR INPUT DATA                                   SSSP
C                                                                SSSP
      CALL OVERLAY(4HSSSP,7,1,6HRECALL)                          CKOUT
      CALL OVERLAY(4HSSSP,7,2,6HRECALL)                          OPTSTG
C                                                                SSSP
C          SET FLAGS FOR CONTINUED ITERATION                     SSSP
C                                                                SSSP
    3 SW(3)    = SW(3) + 1.                                      SSSP
C                                                                SSSP
C          CALL FOR VEHICLE WEIGHTS                              SSSP
C                                                                SSSP
   16 CONTINUE                                                   POW
      IF(SW(13). GT.0..AND. SW(2).GE.1.) GO TO 8                 CKOUT
      IF(SW(13).GE.3.)  GO TO 8                                  CKOUT
      GO TO 9                                                    CKOUT
    8 CALL SUMOUT                                                CKOUT
    9 CONTINUE                                                   CKOUT
C                                                                SSSP
C          CHECK FOR ERROR IN WTVOL ROUTINE                      SSSP
C                                                                SSSP
      IF( EROR.GT.0.5)  GO TO 5                                  SSSP
C                                                                SSSP
C          SET FLAGS FOR NOMINAL RETURN TO PADS                  SSSP
C                                                                SSSP
    6 IPASS    = IPASS + 1                                       SSSP
      CALL WRITMS(3,SSPDAT,1699,1)                               POW
      GO TO 999                                                  SSSP
C                                                                POW
C          OPTIMUM STAGING VELOCITY DETERMINATION                OPTSTG
C                                                                POW
   15 CONTINUE                                                   POW
C                                                                OPTSTG
C          FIRST PASS THRU POWELL                                OPTSTG
      CALL OVERLAY(4HSSSP,7,2,6HRECALL)                          OPTSTG
      DN = 0                                                     OPTSTG
      STEP = 100.                                                OPTSTG
      N = 1                                                      OPTSTG
      PAY = -WPAYLO                                              OPTSTG
      PV = G(TB27(3),TB34(3))                                    OPTSTG
      CALL OPWELL(PV,N,STEP,EPP,PAY)                             OPTSTG
C                                                                OPTSTG
C          CONTINUED ITERATIONS                                  OPTSTG
   21 CONTINUE                                                   OPTSTG
      DELVB = PV                                                 OPTSTG
      DELVO = SV(2) -DELVB                                       OPTSTG
      TB34(3) = F(TB27(3),DELVB)                                 OPTSTG
      TO34(3) = F(TO27(3),DELVO)                                 OPTSTG
      CALL OVERLAY(4HSSSP,7,2,6HRECALL)                          OPTSTG
      CALL SUMOUT                                                OPTSTG
      WRITE(6,94) PAY, PV, WGROSO                                OPTSTG
   94 FORMAT( 9H PAYLOAD= ,E17.8, 19H STAGING VELOCITY =,E17.8,  OPTSTG
     *   16H ORBITER GROSS = ,E17.8)                             OPTSTG
      PV = G(TB27(3),TB34(3))                                    OPTSTG
      PAY = -WPAYLO                                              OPTSTG
      CALL OPWELL(PV,N,STEP,EPP,PAY)                             OPTSTG
      IF(DDN.EQ.-1.)  GO TO 21                                   OPTSTG
      GO TO 17                                                   OPTSTG
C                                                                SSSP
```

```
C          ERROR IN WTVOL                                              SSSP
C                                                                      SSSP
    5 WRITE (6,998)                                                    SSSP
  998 FORMAT (38H CASE ABORTED DUE TO ERROR IN WTVOL    )              SSSP
   12 TRAFLG=2.0                                                       UH
      CALL SIZERR                                                      UH
   10 WRITE(6,11)                                                      UH
   11 FORMAT(36H MAXIMUM ITERATIONS EXCEEDED IN SSSP     )             UH
      GO TO 12                                                         UH
   13 WRITE(6,14)                                                      UH
   14 FORMAT(27H IPASS GREATER THAN IPSMAX     )                       UH
      GO TO 12                                                         UH
C                                                                      SSSP
C          RETURN TO PADS                                              SSSP
C                                                                      SSSP
  999 RETURN                                                           SSSP
      END                                                              SSSP
```

BLØCK
CINPUT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
A		Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH	I	A
					FRENCH	M	ANENGS
					SETO	O	A
					STORE	M	ANENGS
					WTSCH	I	ANENGS
ANTANK		Number of air breathing fuel tanks	/CINPUT/(	2)	STORE	M	ANTANK
					WTSCH	I	ANTANK
ASRATO		Wing aspect ratio	/CINPUT/(	3)	STORE	M	ASRATO
					WTSCH	I	ASRATO
ASWEEP		Wing leading edge sweep angle	/CINPUT/(	4)	STORE	M	ASWEEP
					WTSCH	I	ASWEEP
C		Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
					PRITEQ	I	C
					PRITVA	I	C
					STORE	M	C
					WTSCH	I	C
					WTVOL	O	C
CBBODY		Body width coeff.	/CINPUT/(	305)	PRITVA	I	CBBODY
					STORE	M	CBBODY
					WTSCH	I	CBBODY
CFUEL		Mixture ratio	/CINPUT/(	306)	PRWTSM	M	CFUEL
					STORE	M	CFUEL
					WTSCH	M	CFUEL
CHBODY		Body height or coeff	/CINPUT/(	312)	PRITVA	I	CHBODY
					STORE	M	CHBODY
					WTSCH	I	CHBODY
CLBODY		Body length or coeff	/CINPUT/(	313)	PRITVA	I	CLBODY
					STORE	M	CLBODY
					WTSCH	I	CLBODY
CSBODY		Total body wetted area or coeff	/CINPUT/(	314)	PRITVA	I	CSBODY
					STORE	M	CSBODY
					WTSCH	I	CSBODY
CSFAIR		Fairing planform area or coeff	/CINPUT/(	315)	PRITVA	I	CSFAIR
					STORE	M	CSFAIR
					WTSCH	I	CSFAIR
CSFUTK		Fuel tank surface area or coeff	/CINPUT/(	316)	PRITVA	I	CSFUTK
					STORE	M	CSFUTK
					WTSCH	I	CSFUTK
CSHORZ		Horizontal stabalizer planform area	/CINPUT/(	317)	PRITVA	I	CSHORZ
					STORE	M	CSHORZ
					WTSCH	I	CSHORZ
CSOXTK		Oxidizer tank surface area coeff	/CINPUT/(	318)	PRITVA	I	CSOXTK
					STORE	M	CSOXTK
					WTSCH	I	CSOXTK
CSPLAN		Body planform area or coeff	/CINPUT/(	319)	PRITVA	I	CSPLAN
					STORE	M	CSPLAN
					WTSCH	I	CSPLAN
CSVERT		Vertical fin planform area or coeff	/CINPUT/(	320)	PRITVA	I	CSVERT
					STORE	M	CSVERT
					WTSCH	I	CSVERT
CSWING		Wing planform area	/CINPUT/(	321)	STORE	M	CSWING
CTHRST		Vac. Thrust-to-weight ratio	/CINPUT/(	322)	PRITVA	I	CTHRST
					STORE	M	CTHRST
					WTSCH	I	CTHRST
					WTVOL	M	CTHRST
CTHST2		Secondary propulsion t/w	/CINPUT/(	323)	PRITVA	I	CTHST2
					STORE	M	CTHST2
					WTSCH	I	CTHST2
FXWOVS		Fixed wing loading	/CINPUT/(	329)	STORE	M	FXWOVS
					WTSCH	I	FXWOVS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ISP		Specific impulse	/CINPUT/(	330)	PRWTSM	I	ISP
					STORE	M	ISP
					WTSCH	I	ISP
					WTVOL	O	ISP
ITPS		Thermo protection flag	/CINPUT/(	336)	FRENCH	O	ITPS
					STORE	M	ITPS
					WTSCH	M	ITPS
K		Description not input	/CINPUT/(	337)	PRITEO	I	K
					STORE	M	K
					WTSCH	M	K
LF		Ultimate load factor 1. Thrust buildup 2. Not used 3. Main impulse mass ratio 4. Main impulse reserve 5. Secondary impulse mass ratio 6. Not used	/CINPUT/(	368)	STORE	M	LF
					WTSCH	I	LF
MR		Mass ratio	/CINPUT/(	369)	PRWTSM	I	MR
					SOLVE	I	MR
					STORE	M	MR
					WTSCH	M	MR
NCREW		Number of crew members	/CINPUT/(	375)	PRITVA	I	NCREW
					STORE	M	NCREW
					WTSCH	I	NCREW
NENGS		Total number engines per stage	/CINPUT/(	376)	PRITVA	I	NENGS
					STORE	M	NENGS
					WTSCH	I	NENGS
NLISTO		Namelist output flag	/CINPUT/(	377)	STORE	M	NLISTO
					WTSCH	I	NLISTO
NPASS		Number of passengers	/CINPUT/(	378)	STORE	M	NPASS
					WTSCH	I	NPASS
NWL		Wing loading flag	/CINPUT/(	379)	FRENCH	O	NWL
					STORE	M	NWL
					WTSCH	M	NWL
PCHAM		Main rocket engine chamber pressure	/CINPUT/(	380)	STORE	M	PCHAM
					WTSCH	I	PCHAM
Q		Maximum dynamic pressure	/CINPUT/(	381)	WTSCH	M	Q
RHOFU		Fuel density	/CINPUT/(	382)	PRITVA	I	RHOFU
					STORE	M	RHOFU
					WTSCH	I	RHOFU
RHOFU2		Secondary fuel desnity	/CINPUT/(	383)	PRITVA	I	RHOFU2
					STORE	M	RHOFU2
					WTSCH	I	RHOFU2
RHOX		Oxidizer density	/CINPUT/(	384)	PRITVA	I	RHOX
					STORE	M	RHOX
					WTSCH	I	RHOX
RHOX2		Secondary oxidizer density	/CINPUT/(	385)	PRITVA	I	RHOX2
					STORE	M	RHOX2
					WTSCH	I	RHOX2
SBODY		Total body wetted area	/CINPUT/(	386)	PROTHR	I	SBODY
					TAMPER	I	SBODY
					WTSCH	M	SBODY
TOL		Gross weight iteration tolerance	/CINPUT/(	387)	SOLVE	M	TOL
TOVERC		Wing thickness over choord ratio	/CINPUT/(	388)	PROTHR	I	TOVERC
					STORE	M	TOVERC
					WTSCH	I	TOVERC
TPRATO		Wing taper ratio	/CINPUT/(	389)	STORE	M	TPRATO
					WTSCH	I	TPRATO
TYTAIL		Description not input	/CINPUT/(	390)	STORE	M	TYTAIL
VBODY		Total body volume	/CINPUT/(	391)	PRINTV	M	VBODY
					SOLVE	M	VBODY
					STORE	M	VBODY
					TAMPER	I	VBODY
					WTSCH	M	VBODY
					WTVOL	I	VBODY

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WGROSS		Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
					PRWTSM	I	WGROSS
					SOLVE	M	WGROSS
					STORE	M	WGROSS
					TAMPER	I	WGROSS
					WTSCH	M	WGROSS
					WTVOL	I	WGROSS

BLØCK
DATA2X

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ALD		Booster sub-sonic l/d	/DATA2X/(	1)	SSSP	D	A
					VEHDF	M	A
					VEHDF	I	ALD
FBPAR		Estimate of slope for booster cruise adjustment if woreq 70 or wporeq 70	/DATA2X/(	2)	VEHDF	I	FBPAR
IDVEL		Total ideal velocity estimate to parking orbit insertion	/DATA2X/(	3)	VEHDF	I	IDVEL
ISLB		Booster sea level specific impulse	/DATA2X/(	4)			
IVACB		Description not input	/DATA2X/(	6)	VEHDF	I	IVACB
IVACO		Description not input	/DATA2X/(	7)	VEHDF	I	IVACO
ISLO		Orbiter sea level specific impulse	/DATA2X/(	8)	VEHDF	I	PERIS
QMXX		Description not input	/DATA2X/(	9)	VEHDF	I	QMXX
QMXS		Description not input	/DATA2X/(	10)	VEHDF	I	QMXS
SFC		Description not input	/DATA2X/(	11)	VEHDF	I	SFC
SLVOUT		Description not input	/DATA2X/(	12)	VEHDF	I	SLVOU
COPIES		Description not input	/DATA2X/(	13)	VEHDF	I	COPIE
SYNIT		Description not input	/DATA2X/(	14)	VEHDF	I	SYNIT
IVACB		Booster bac. Isp	/DATA2X/(	15)	VEHDF	I	TFCTR
TFCTRO		Description not input	/DATA2X/(	16)	VEHDF	I	TFCTR
TOLMU		Description not input	/DATA2X/(	17)	VEHDF	I	TOLMU
TOLTW		Description not input	/DATA2X/(	18)	VEHDF	I	TOLTW
IVACO		Orbiter vac. Isp	/DATA2X/(	19)	VEHDF	I	TRATI
TWLO		Description not input	/DATA2X/(	20)	VEHDF	I	TWLO
TWLOI		Description not input	/DATA2X/(	21)	VEHDF	I	TWLOI
WTOUT		Description not input	/DATA2X/(	22)	VEHDF	I	WTOUT
FIRE		Description not input	/DATA2X/(	23)	VEHDF	I	FIRE
BOOTW		Description not input	/DATA2X/(	24)	VEHDF	I	BOOTW
VCRUSE		Description not input	/DATA2X/(	25)	VEHDF	I	VCRUS
PERISP		Booster effective isp estimating parameter	/DATA2X/(	26)	VEHDF	I	NXFOB
QMAX		Estimate of max q during ascent used for sizing vehicle	/DATA2X/(	27)			
QMXS		Slope used for max-q adjustment when t/w l.o. Varies during synthesis iterations	/DATA2X/(	28)			
SFC		Specific fuel consumption of booster air breathers	/DATA2X/(	29)	VEHDF	I	CLVG
SLVOUT		Output flag for sizing data	/DATA2X/(	30)	VEHDF	I	DRNG
COPIES		Number of copies of summary sheet to be output	/DATA2X/(	31)	VEHDF	I	SOLID
SYNIT		Number of allowable synthesis iterat -ions (max = 6)	/DATA2X/(	32)	VEHDF	I	AS
TFCTRB		Booster thrust multiplier for ascent	/DATA2X/(	33)	VEHDF	I	BS
SISP		Description not input	/DATA2X/(	34)	VEHDF	I	SISP
SINERT		Description not input	/DATA2X/(	35)	VEHDF	I	SINER
SAE		Description not input	/DATA2X/(	36)	VEHDF	I	SAE
TFCTRO		Orbiter thrust multiplier for ascent	/DATA2X/(	37)	VEHDF	I	TSBO
FLYBCK		Description not input	/DATA2X/(	38)	VEHDF	I	FLYBC
WPOREQ		Description not input	/DATA2X/(	39)	VEHDF	I	WPORE
WOREQ		Description not input	/DATA2X/(	40)	VEHDF	I	WOREQ
GWREQ		Description not input	/DATA2X/(	41)	VEHDF	I	GWREQ
FBFUEL		Description not input	/DATA2X/(	42)	VEHDF	I	FBFUI
CA		Description not input	/DATA2X/(	43)	VEHDF	I	CA
TOLMU		Orbiter mass ratio tolerance input	/DATA2X/(	44)	VEHDF	I	CB
TOLTW		Tolerance on twlo iteration	/DATA2X/(	45)	VEHDF	I	WFLYI
TRATIO		Ratio of booster to orbiter engine thrust (vac)	/DATA2X/(	46)	VEHDF	I	RT

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
TWLO		Desired value of l.o. T/w	/DATA2X/(	47)	VEHDF	I	R1
TWLO1		Maximum number of iterations to obtain twlo	/DATA2X/(	48)	VEHDF	I	R3
WTOUT		Output print flag	/DATA2X/(	49)	VEHDF	I	SFC1
FIRE		Ascent burn sequence flag 1= simultaneous stage burn 2= sequential stage burn	/DATA2X/(	50)	VEHDF	I	SFC2
SFC3		Description not input	/DATA2X/(	51)	VEHDF	I	SFC3
VCRUSE		Booster flyback cruise velocity	/DATA2X/(	52)	VEHDF	I	ALD1
NXFOB		Flag for crossfeed of propellants from booster tanks to orbiter engines at lift-off if fire =1	/DATA2X/(	53)	VEHDF	I	ALD2
PRNTX		Data/ print flag	/DATA2X/(	54)	VEHDF	I	ALD3
VFLY1		Description not input	/DATA2X/(	55)	VEHDF	I	VFLY1
VFLY2		Description not input	/DATA2X/(	56)	VEHDF	I	VFLY2
VFLY3		Description not input	/DATA2X/(	57)	VEHDF	I	VFLY3
SOLID		Number of solid motors also flag for solid motor option	/DATA2X/(	58)			
AS		Solid motor thrust curve intercept	/DATA2X/(	59)	HUNT	I	RVAR
BS		Solid motor thrust slope	/DATA2X/(	60)	HUNT	I	PNDX
SISP		Solid motor vacuum isp	/DATA2X/(	61)			
SINERT		Solid motor inert weight	/DATA2X/(	62)			
SAE		Solid motor exit area	/DATA2X/(	63)			
TSBO		Solid rocket burn time	/DATA2X/(	64)			
FBFUEL		Flyback cruise calculation flag 1= parametric range data 2= staging- q function 3= constant range 4= ballistic impact range 5= entry trajectory simulation	/DATA2X/(	65)			
CA		Per cent booster weight for cruise back leg no. 1	/DATA2X/(	66)			
CB		Per cent booster weight for cruise back leg no. 2	/DATA2X/(	67)			
WFLYX		Addative booster weight for flyback	/DATA2X/(	68)			
RT		Transition range increment for flyback calculations	/DATA2X/(	69)			
R1		Idle descent range increment for flyback calculations	/DATA2X/(	70)	HUNT	I	BLOW
R3		Final descent flyback range incremnt	/DATA2X/(	71)			
SFC1		Specific fuel consumption	/DATA2X/(	72)			
SFC2		Specific fuel consumption	/DATA2X/(	73)			
SFC3		Specific fuel consumption	/DATA2X/(	74)			
ALD3		Booster sub-sonic l/d	/DATA2X/(	77)			
BUPP		Description not input	/DATA2X/(	80)	HUNT	I	BUPP
STEP		Description not input	/DATA2X/(	90)	HUNT SSSP	M M	STEP STEP
PAYX		Description not input	/DATA2X/(	101)	HUNT	I	PAYX
VFLY1		Booster flyback cruise velocity pse1	/DATA2X/(	78)			
VFLY2		Booster flyback cruise velocity pse2	/DATA2X/(	79)			
VFLY3		Booster flyback cruise velocity pse3	/DATA2X/(	80)			

1417

BLØCK
EMS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR CODE	SUBROUTINE USAGE VAR
WORBIT		Weight in orbit	/EMS	/(1)	PRWTSM O TAMPER I	WORBIT WORBIT
WETURN		Weight at return point	/EMS	/(2)	PRWTSM O TAMPER I	WETURN WETURN
WENTRY		Entry weight	/EMS	/(3)	PRWTSM O TAMPER I	WENTRY WENTRY
WLAND		Landing weight	/EMS	/(4)	PRWTSM O TAMPER I	WLAND WLAND

BLØCK
JUMPY

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
JUMP		Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH	O	JUMP
					PRINTW	I	JUMP
					PRITVA	I	JUMP
					PRWTSM	M	JUMP
					WTSCH	I	JUMP
					WTVOL	M	JUMP
WBOO		Booster gross weight	/JUMPY /(	3)	PRINTW	I	WBOO
					TAMPER	I	WBOO
					WTVOL	O	WBOO

1422

BLØCK
ØRBINX

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T02		Stored orbiter value of cbbody	/ORBINX/(	2)	STORE	M	T02
T04		Stored orbiter value of cfuel(I)	/ORBINX/(	4)	STORE	M	T04
T05		Stored orbiter value of chbody	/ORBINX/(	10)	STORE	M	T05
T06		Stored orbiter value of clbody	/ORBINX/(	11)	STORE	M	T06
T07		Stored orbiter value of csbody	/ORBINX/(	12)	STORE	M	T07
T010		Stored orbiter value of csfair	/ORBINX/(	15)	STORE	M	T010
T011		Stored orbiter value of csfutk	/ORBINX/(	16)	STORE	M	T011
T012		Stored orbiter value of csoxtk	/ORBINX/(	17)	STORE	M	T012
T013		Stored orbiter value of cshorz	/ORBINX/(	18)	STORE	M	T013
T015		Stored orbiter value of csplam	/ORBINX/(	20)	STORE	M	T015
T016		Stored orbiter value of csvert	/ORBINX/(	21)	STORE	M	T016
T017		Stored orbiter value of cswing	/ORBINX/(	22)	STORE	M	T017
T018		Stored orbiter value of cthrst	/ORBINX/(	23)	STORE	M	T018
					WTVOL	M	T018
T019		Stored orbiter value of cthstz	/ORBINX/(	24)	STORE	M	T019
T027		Stored orbiter value of isp(I)	/ORBINX/(	41)	SIZEMA	I	T027
					SSSP	I	T027
					STORE	M	T027
					SUMOUT	I	T027
					VEHDF	M	T027
					WTVOL	I	T027
T034		Stored orbiter value of mr(I)	/ORBINX/(	53)	ITER8	M	T034
					SSSP	O	T034
					STORE	M	T034
					TAMPER	I	T034
					VEHDF	O	T034
					WTVOL	M	T034
T035		Stored orbiter value of ncrew	/ORBINX/(	59)	STORE	M	T035
T036		Stored orbiter value of nengs	/ORBINX/(	60)	STORE	M	T036
					SUMOUT	I	T036
					TAMPER	I	T036
					THRUST	I	T036
					WTVOL	I	T036
T038		Stored orbiter value of npass	/ORBINX/(	62)	STORE	M	T038
T041		Stored orbiter value of rhofu	/ORBINX/(	65)	STORE	M	T041
T042		Stored orbiter value of rhofu2	/ORBINX/(	66)	STORE	M	T042
T043		Stored orbiter value of rhox	/ORBINX/(	67)	STORE	M	T043
T044		Description not input	/ORBINX/(	68)	STORE	M	T044
T045		Stored orbiter value of swing	/ORBINX/(	69)	STORE	M	T045
					SUMOUT	I	T045
					TAMPER	I	T045
T047		Stored orbiter value of toverc	/ORBINX/(	71)	STORE	M	T047
T051		Stored orbiter value of vbody	/ORBINX/(	102)	STORE	M	T051
					WTVOL	O	T051
T052		Stored orbiter value of vfutk	/ORBINX/(	103)	STORE	O	T052
T053		Stored orbiter value of vfutk2	/ORBINX/(	104)	STORE	M	T053
T054		Stored orbiter value of voxtk	/ORBINX/(	105)	STORE	O	T05
T055		Stored orbiter value of voxtk2	/ORBINX/(	106)	STORE	M	T05
T056		Stored orbiter value of wgross	/ORBINX/(	107)	STORE	M	T05
					WTVOL	M	T05
SK0		Working name for input k-array orbiter volume scaling coeff	/ORBINX/(	114)	STORE	M	T06
SK0		Description not input	/ORBINX/(	115)	STORE	M	SK0
SC0		Working name for input c-array orbiter scaling coefficients	/ORBINX/(	144)			

1425

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SCO		Description not input	/ORBINX/(	145)	STORE	M	SCO
					TAMPER	I	SCO
					THRUST	M	SCO
					WTVOL	M	SCO
OWSAVE		Description not input	/ORBINX/(	445)	STORE	M	OWSAVE
					TAMPER	I	OWSAVE
T059		Description not input	/ORBINX/(	455)	STORE	M	T059
T060		Description not input	/ORBINX/(	456)	STORE	M	T060
T061		Description not input	/ORBINX/(	457)	STORE	M	T061
T062		Description not input	/ORBINX/(	458)	STORE	M	T062
T063		Description not input	/ORBINX/(	459)	STORE	M	T063
T064		Description not input	/ORBINX/(	460)	STORE	M	T064
T065		Description not input	/ORBINX/(	461)	STORE	M	T065
T067		Description not input	/ORBINX/(	462)	STORE	M	T067
T068		Description not input	/ORBINX/(	463)	STORE	M	T068
T069		Description not input	/ORBINX/(	464)	STORE	M	T069
T070		Description not input	/ORBINX/(	465)	STORE	M	T070
T071		Description not input	/ORBINX/(	466)	STORE	M	T071
T072		Description not input	/ORBINX/(	467)	STORE	M	T072
T073		Description not input	/ORBINX/(	468)	STORE	M	T073
T074		Description not input	/ORBINX/(	469)	STORE	M	T074
T075		Description not input	/ORBINX/(	470)	STORE	M	T075
T076		Description not input	/ORBINX/(	471)	STORE	M	T076
T077		Description not input	/ORBINX/(	472)	STORE	M	T077
T078		Description not input	/ORBINX/(	473)	STORE	M	T078
T079		Description not input	/ORBINX/(	474)	STORE	M	T079
T080		Description not input	/ORBINX/(	475)	STORE	M	T080
T081		Description not input	/ORBINX/(	476)	STORE	M	T081
T082		Description not input	/ORBINX/(	477)	STORE	M	T082
T083		Description not input	/ORBINX/(	478)	STORE	M	T083
T084		Description not input	/ORBINX/(	479)	STORE	M	T084

BLØCK
ØRBINY

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USA SUBR	CODE	VA
A		Description not input	/ORBINY/(	1)	HUNT	M	A
					SSSP	I	SSPD
TB2		Stored booster value of cbbody	/ORBINY/(	2)	STORE	M	TB2
TB59		Stored booster value of lf	/ORBINY/(	4)	STORE	M	TB4
TB4		Stored booster value of cfuel(I)	/ORBINY/(	4)			
TB5		Stored booster value of chbody	/ORBINY/(	10)	STORE	M	TB5
TB6		Stored booster value of clbody	/ORBINY/(	11)	STORE	M	TB6
TB7		Description not input	/ORBINY/(	12)	STORE	M	TB7
TB10		Stored booster value of csfair	/ORBINY/(	15)	STORE	M	TB10
TB11		Stored booster value of csfutk	/ORBINY/(	16)	STORE	M	TB11
TB12		Stored booster value of csoxtk	/ORBINY/(	17)	STORE	M	TB12
TB13		Stored booster value of cshorz	/ORBINY/(	18)	STORE	M	TB13
TB15		Stored booster value of csplan	/ORBINY/(	20)	STORE	M	TB15
TB16		Stored booster value of csvert	/ORBINY/(	21)	STORE	M	TB16
TB17		Stored booster value of cswing	/ORBINY/(	22)	STORE	M	TB17
TB18		Stored booster value of cthrst	/ORBINY/(	23)	STORE	M	TB18
					WTVOL	O	TB18
TB19		Stored booster value of cthst2	/ORBINY/(	24)	STORE	M	TB19
TB27		Stored booster value of isp(I)	/ORBINY/(	41)	SIZEMR	I	TB27
					SSSP	I	TB27
					STORE	M	TB27
					SUMOUT	I	TB27
					TAMPER	I	TB27
					VEHOF	M	TB27
					WTVOL	I	TB27
TB34		Stored booster value of mr(I)	/ORBINY/(	53)	FLYBKP	I	TB34
					ITER8	O	TB34
					SSSP	M	TB34
					STORE	M	TB34
					SUMOUT	I	TB34
					TAMPER	I	TB34
					VEHOF	I	TB34
					WTVOL	M	TB34
TB35		Stored booster value of ncrew	/ORBINY/(	59)	STORE	M	TB35
TB36		Stored booster value of nengs	/ORBINY/(	60)	STORE	M	TB36
					SUMOUT	I	TB36
					TAMPER	I	TB36
					THRUST	I	TB36
					WTVOL	I	TB36
TB38		Stored booster value of npass	/ORBINY/(	62)	STORE	M	TB38
TB41		Stored booster value of rhofu	/ORBINY/(	65)	STORE	M	TB41
TB42		Stored booster value of rhofu2	/ORBINY/(	66)	STORE	M	TB42
TB43		Stored booster value of rhox	/ORBINY/(	67)	STORE	M	TB43
TB44		Stored booster value of rhox2	/ORBINY/(	68)	STORE	M	TB44
TB45		Stored booster value of swing	/ORBINY/(	69)	STORE	M	TB45
					SUMOUT	I	TB45
					TAMPER	I	TB45
TB47		Stored booster value of toverc	/ORBINY/(	71)	STORE	M	TB47
TB51		Stored booster value of vbody	/ORBINY/(	102)	STORE	M	TB51
					WTVOL	O	TB51
TB51		Stored booster value of vfutk	/ORBINY/(	103)	STORE	O	TB52
TB53		Stored booster value of vfutk2	/ORBINY/(	104)	STORE	M	TB53
TB54		Stored booster value of voxtk	/ORBINY/(	105)	STORE	O	TB54
TB55		Stored booster value of voxtk2	/ORBINY/(	106)	STORE	M	TB55
TB56		Stored booster value of wgross	/ORBINY/(	107)	STORE	M	TB56
					WTVOL	M	TB56

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SKB		Working name for input k-array booster volume scaling coeff	/ORBINY/(	114)	STORE	M	SKB
SCB		Working name for input c-array booster scaling coefficients	/ORBINY/(	144)	FLYBKP	M	SCB
					STORE	M	SCB
					SUMOUT	I	SCB
					TAMPER	I	SCB
					THRUST	M	SCB
					VEHOF	I	SCB
					WTVOL	M	SCB
BWSAVE		Description not input	/ORBINY/(	444)	STORE	M	BWSAVE
TB59		Description not input	/ORBINY/(	454)	STORE	M	TB59
TB60		Description not input	/ORBINY/(	455)	STORE	M	TB60
TB61		Description not input	/ORBINY/(	456)	STORE	M	TB61
TB62		Description not input	/ORBINY/(	457)	STORE	M	TB62
TB63		Description not input	/ORBINY/(	458)	STORE	M	TB63
TB64		Description not input	/ORBINY/(	459)	STORE	M	TB64
TB65		Description not input	/ORBINY/(	460)	STORE	M	TB65
TB66		Description not input	/ORBINY/(	461)	STORE	M	TB66
TB67		Description not input	/ORBINY/(	462)	STORE	M	TB67
TB68		Description not input	/ORBINY/(	463)	STORE	M	TB68
TB69		Description not input	/ORBINY/(	464)	STORE	M	TB69
TB70		Description not input	/ORBINY/(	465)	STORE	M	TB70
TB71		Description not input	/ORBINY/(	466)	STORE	M	TB71
TB72		Description not input	/ORBINY/(	467)	STORE	M	TB72
TB73		Description not input	/ORBINY/(	468)	STORE	M	TB73
TB74		Description not input	/ORBINY/(	469)	STORE	M	TB74
TB75		Description not input	/ORBINY/(	470)	STORE	M	TB75
TB76		Description not input	/ORBINY/(	471)	STORE	M	TB76
TB77		Description not input	/ORBINY/(	472)	STORE	M	TB77
TB78		Description not input	/ORBINY/(	473)	STORE	M	TB78
TB79		Description not input	/ORBINY/(	474)	STORE	M	TB79
TB80		Description not input	/ORBINY/(	475)	STORE	M	TB80
TB81		Description not input	/ORBINY/(	476)	STORE	M	TB81
TB82		Description not input	/ORBINY/(	477)	STORE	M	TB82
TB83		Description not input	/ORBINY/(	478)	STORE	M	TB83
TB84		Description not input	/ORBINY/(	479)	STORE	M	TB84

1429

BLØCK
PØ

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
MPRNT		Description not input	/PO	/(1)	FRENCH	M	MPRNT
					VEHOF	I	MPRNT

BLØCK
PRESET

1432

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
PRESET		Description ~~[illegible]~~ PRESETS FOR DATA2X	/PRESET/(	1)	SSSP VEHDF	D I	PRESET PRESET

BLØCK
SUMVW

1434

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WOP		Orbit maneuvering propellant weight	/SUMVW /(	1)	SUMOUT TAMPER	I O	WOP WOP
WABFUB		Booster flyback fuel required	/SUMVW /(	2)	SUMOUT TAMPER	I M	WABFUB WABFUB
WFUOXB		Propellant wt. Less fpr -booster	/SUMVW /(	3)	SUMOUT TAMPER	I M	WFUOXB WFUOXB
WDRYB		Stage dry weight - booster	/SUMVW /(	4)	SUMOUT TAMPER	I O	WDRYB WDRYB
WGROSB		Booster gross weight	/SUMVW /(	5)	SUMOUT TAMPER	I M	WGROSB WGROSB
WOTHB		Misc. Weight - booster	/SUMVW /(	6)	SUMOUT TAMPER	I O	WOTHB WOTHB
VFUTKB		Total volume of fuel tank - booster	/SUMVW /(	7)	SUMOUT TAMPER	I O	VFUTKB VFUTKB
VOXTKB		Booster oxidizer tank volume	/SUMVW /(	8)	SUMOUT TAMPER	I O	VOXTKB VOXTKB
VOTHB		Misc. Booster volume	/SUMVW /(	9)	SUMOUT TAMPER	I O	VOTHB VOTHB
VBODYB		Booster body volume	/SUMVW /(	10)	SUMOUT TAMPER	I O	VBODYB VBODYB
VABFUB		Volume of booster propellant tanks	/SUMVW /(	11)	TAMPER	O	VABFUB
LBODYB		Description not input	/SUMVW /(	12)	SUMOUT TAMPER	I O	LBODYB LBODYB
SBODYB		Total body wetted area- booster	/SUMVW /(	13)	SUMOUT TAMPER	I O	SBODYB SBODYB
SPLANB		Booster body planform area	/SUMVW /(	14)	SUMOUT TAMPER	I O	SPLANB SPLANB
WOVRSB		Booster wing loading	/SUMVW /(	15)	SUMOUT TAMPER	I O	WOVRSB WOVRSB
WPAYLO		Payload weight	/SUMVW /(	16)	SSSP SUMOUT TAMPER	I I M	WPAYLO WPAYLO WPAYLO
WDRYO		Stage dry weight -orbiter	/SUMVW /(	17)	SUMOUT TAMPER	I O	WDRYO WDRYO
WOTHO		Misc. Weight - orbiter	/SUMVW /(	18)	SUMOUT TAMPER	I O	WOTHO WOTHO
WABFUO		Orbiter flyback fuel required	/SUMVW /(	19)	SUMOUT TAMPER	I M	WABFUO WABFUO
VFUTKO		Total volume of fuel tank - orbiter	/SUMVW /(	20)	SUMOUT TAMPER	I O	VFUTKO VFUTKO
VOXTKO		Orbiter oxidizer tank volume	/SUMVW /(	21)	SUMOUT TAMPER	I O	VOXTKO VOXTKO
VCARGO		Volume of cargo	/SUMVW /(	22)	SUMOUT TAMPER	I O	VCARO VCARO
VOTHO		Misc. Orbiter volume	/SUMVW /(	23)	SUMOUT TAMPER	I O	VOTHO VOTHO
VBODYO		Orbiter body volume	/SUMVW /(	24)	SUMOUT TAMPER	I O	VBODYO VBODYO
LBODYO		Description not input	/SUMVW /(	25)	SUMOUT TAMPER	I O	LBODYO LBODYO
SBODYO		Total body wetted area- orbiter	/SUMVW /(	26)	SUMOUT TAMPER	I O	SBODYO SBODYO
SPLANO		Orbiter body planform area	/SUMVW /(	27)	SUMOUT TAMPER	I O	SPLANO SPLANO
WOVRSO		Orbiter wing loading	/SUMVW /(	28)	SUMOUT TAMPER	I O	WOVRSO WOVRSO
WORBTO		In-orbit weight - orbiter	/SUMVW /(	29)	SUMOUT TAMPER	I O	WORBTO WORBTO
WORBTB		In-orbit weight - booster	/SUMVW /(	30)	TAMPER	O	WORBTB

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WRTRNO		Entry weight- orbiter	/SUMVW /(	31)	SUMOUT TAMPER	I O	WRTRNO WRTRNO
WRTRNB		Entry weight- booster	/SUMVW /(	32)	SUMOUT TAMPER	I O	WRTRNB WRTRNB
WENTRO		Orbiter entry weight	/SUMVW /(	33)	SUMOUT TAMPER	I O	WENTRO WENTRO
WENTRB		Booster entry weight	/SUMVW /(	34)	SUMOUT TAMPER	I O	WENTRB WENTRB
WLANDB		Landing weight - booster	/SUMVW /(	35)	SUMOUT TAMPER	I O	WLANDB WLANDB
WLANDO		Landing weight - orbiter	/SUMVW /(	36)	SUMOUT TAMPER	I O	WLANDO WLANDO
WCONTO		Contingency and growth weight-orbitr	/SUMVW /(	37)	SUMOUT TAMPER	I O	WCONTO WCONTO
WCONTB		Contingency and growth weight-boostr	/SUMVW /(	38)	SUMOUT TAMPER	I O	WCONTB WCONTB

1436

BLØCK
TAMP

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WGROSO		Orbiter gross weight	/TAMP	/(1)	SSSP	I	WGROSO
					SUMOUT	I	WGROSO
					WTVOL	M	WGROSO
OTTOT		Total orbiter thrust	/TAMP	/(2)	SUMOUT	I	OTTOT
					TAMPER	I	OTTOT
					WTVOL	M	OTTOT
WFUOXO		Propellant wt. Less fpr -orbiter	/TAMP	/(3)	SUMOUT	I	WFUOXO
					TAMPER	I	WFUOXO
					WTVOL	M	WFUOXO
TBTO		Description not input	/TAMP	/(4)	WTVOL	M	TBTO
BTTOT		Total booster weight flow	/TAMP	/(5)	SUMOUT	I	BTTOT
					TAMPER	I	BTTOT
					WTVOL	O	BTTOT

BLØCK
TRUST

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR CODE	VAR
FVACO		Orbiter vacuum thrust (lb)	/THRST /(	1)		

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
FVACO		Description not input	/TRUST /(	1)	THRUST	M	FVACO
					WTSCH	I	FVACO
FSLO		Orbiter sea level thrust (lb)	/TRUST /(	2)	THRUST	M	FSLO
FVACLO		Total vacuum lift-off thrust (lb)	/TRUST /(	3)	THRUST	M	FVACLO
FVACS		Solid motor total vacuum thrust (lb)	/TRUST /(	4)	THRUST	M	FVACS
FVACB		Booster vacuum thrust (lb)	/TRUST /(	5)	THRUST	M	FVACB
					WTSCH	M	FVACB
FSLB		Booster sea level thrust (lb)	/TRUST /(	6)	THRUST	M	FSLB
FSLLO		Sea level lift-off thrust (lb)	/TRUST /(	7)	THRUST	M	FSLLO
FSLS		Solid motor sea level thrust (lb)	/TRUST /(	8)	THRUST	M	FSLS

BLØCK
VØLCAL

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
BBODY		Body width used by set0 to set common to zero	/VOLCAL/(	1)	PROTHR SETO WTSCH	I O M	BBODY D BBODY
CROOT		Ing root chord	/VOLCAL/(	2)	PROTHR WTSCH	I M	CROOT CROOT
CSPAN		Structural span along 0.5 chord	/VOLCAL/(	3)	PROTHR WTSCH	I M	CSPAN CSPAN
CTIP		Wing tip chord	/VOLCAL/(	4)	WTSCH	O	CTIP
GAL		Total gallons of fuel	/VOLCAL/(	5)	WTSCH	M	GAL
GSPAN		Geometric wing span	/VOLCAL/(	6)	WTSCH	M	GSPAN
HBODY		Body height	/VOLCAL/(	7)	PROTHR WTSCH	I O	HBODY HBODY
LBODY		Body length	/VOLCAL/(	8)	PROTHR TAMPER WTSCH	I I M	LBODY LBODY LBODY
RTOD		Deg to rad conversion	/VOLCAL/(	9)	WTSCH	M	RTOD
SFAIR		Total fairing or shroud surface area	/VOLCAL/(	10)	WTSCH	M	SFAIR
SFUTK		Total fuel tank wetted area	/VOLCAL/(	11)	PROTHR WTSCH	I M	SFUTK SFUTK
SHORZ		Horizontal stabilizer planform area	/VOLCAL/(	12)	PROTHR WTSCH	I M	SHORZ SHORZ
SOXTK		Total oxidizer tank wetted area	/VOLCAL/(	13)	PROTHR WTSCH	I M	SOXTK SOXTK
SPLAN		Body planform area	/VOLCAL/(	14)	PROTHR TAMPER WTSCH WTVOL	I I O I	SPLAN SPLAN SPLAN SPLAN
STPS		Total thermal protection system surface area	/VOLCAL/(	15)	WTSCH	M	STPS
SVERT		Vertical fin planform area	/VOLCAL/(	16)	PROTHR WTSCH	I M	SVERT SVERT
SWING		Gross wing area	/VOLCAL/(	17)	PROTHR STORE WTSCH	I M M	SWING SWING SWING
SXPOS		Exposed wing area	/VOLCAL/(	18)	PROTHR WTSCH	I O	SXPOS SXPOS
TDEL		Gimbal system delivered torque	/VOLCAL/(	19)	WTSCH	M	TDEL
TROOT		Theoritical root thickness	/VOLCAL/(	20)	WTSCH	M	TROOT
TTOT		Total stage vac. Thrust	/VOLCAL/(	21)	PRITVA STORE WTSCH WTVOL	I I M I	TTOT TTOT TTOT TTOT
TTOT2		Total stage vac. Secondary thrust	/VOLCAL/(	22)	PRITVA WTSCH	I M	TTOT2 TTOT2
TTOTAL		Total stage vac. Thrust / 1,000,000	/VOLCAL/(	23)	WTSCH	O	TTOTAL
VBODYA		Total body volume less structure	/VOLCAL/(	24)	WTSCH	M	VBODYA
VBODY1		Vbody to - 1/3 power	/VOLCAL/(	25)	WTSCH WTVOL	M M	VBODY1 VBODY1
VBODY2		Vbody to - 2/3 power	/VOLCAL/(	26)	WTSCH WTVOL	M M	VBODY2 VBODY2
VCARGO		Volume of cargo bay	/VOLCAL/(	27)	PRINTV TAMPER WTSCH	I I M	VCARGO VCARGO VCARGO
VCREW		Volume of crew compartment	/VOLCAL/(	28)	PRINTV WTSCH	I M	VCREW VCREW
VFUTK		Total volume of fuel tank	/VOLCAL/(	29)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK VFUTK VFUTK VFUTK

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
VFUTK2		Total volume of secondary fuel tank	/VOLCAL/(	30)	PRINTV	I	VFUTK2
					STORE	M	VFUTK2
					TAMPER	I	VFUTK2
					WTSCH	M	VFUTK2
VINSTK		Total tank insulation volume	/VOLCAL/(	31)	PRINTV	I	VINSTK
					WTSCH	M	VINSTK
VLGBAY		Volume of recovery system bay	/VOLCAL/(	32)	PRINTV	I	VLGBAY
					WTSCH	M	VLGBAY
VOTHER		Misc. And unused volume	/VOLCAL/(	32)			
VOTHER		Description not input	/VOLCAL/(	33)	PRINTV	I	VOTHER
					WTSCH	M	VOTHER
VOXTK		Total volume of oxidizer tank	/VOLCAL/(	34)	PRINTV	I	VOXTK
					STORE	M	VOXTK
					TAMPER	I	VOXTK
					WTSCH	M	VOXTK
VOXTK2		Total volume of secondary oxidizer tank	/VOLCAL/(	35)	PRINTV	I	VOXTK2
					STORE	M	VOXTK2
					TAMPER	I	VOXTK2
					WTSCH	M	VOXTK2
VPROP		Volume of propulsion bay	/VOLCAL/(	36)	PRINTV	I	VPROP
					WTSCH	M	VPROP
VSTRUC		Volume of basic structure	/VOLCAL/(	37)	PRINTV	I	VSTRUC
					WTSCH	M	VSTRUC

1444

BLØCK
WTCALC

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
C		Airbreathing fuel system weight used by set0 to set common to zero	/WTCALC/(	1)	SETO WTSCH	O M	C ABFSYS
WABFTK		Weight of air breathing propulsion system tanks	/WTCALC/(	2)	PRINTW WTSCH	I M	WABFTK WABFTK
WABFU		Weight of jp fuel	/WTCALC/(	3)	PRINTW PRWTSM TAMPER WTSCH	I M I M	WABFU WABFU WABFU WABFU
WABPR		Weight of air breathing engines	/WTCALC/(	4)	PRINTW WTSCH	I M	WABPR WABPR
WACRES		Weight of attitude control fuel reserve	/WTCALC/(	5)	PRINTW STORE WTSCH	I M M	WACRES WACRES WACRES
WACS		Weight of attitude control system	/WTCALC/(	6)	PRINTW WTSCH	I M	WACS WACS
WACSFO		Weight of attitude control fuel plus oxidizer	/WTCALC/(	7)	PRINTW STORE WTSCH	I M M	WACSFO WACSFO WACSFO
WACSTK		Weight of attitude control tankage	/WTCALC/(	8)	PRINTW WTSCH	I M	WACSTK WACSTK
WAERO		Weight of aerodynamic controls	/WTCALC/(	9)	PRINTW WTSCH	I M	WAERO WAERO
WAUXT		Weight of separation system	/WTCALC/(	10)	PRINTW STORE WTSCH	I I M	WAUXT WAUXT WAUXT
WBASIC		Total weight of basic body	/WTCALC/(	11)	PRINTW PROTHR WTSCH	I I M	WBASIC WBASIC WBASIC
WBODY		Total weight of body group	/WTCALC/(	12)	PRINTW WTSCH	I M	WBODY WBODY
WBPUMP		Weight of boost and transfer pumps	/WTCALC/(	13)	WTSCH	M	WBPUMP
WCARGO		Payload weight or cargo	/WTCALC/(	14)	PRINTW WTSCH	I M	WCARGO WCARGO
WCOMM		Communication system weight	/WTCALC/(	15)	PRINTW WTSCH	I M	WCOMM WCOMM
WCONT		Contingency and growth weight	/WTCALC/(	16)	PRINTW TAMPER WTSCH	I I M	WCONT WCONT WCONT
WCOVER		Total weight of thermal protection system cover panels	/WTCALC/(	17)	PRINTW WTSCH	I M	WCOVER WCOVER
WDECAY		Thrust decay propellant weight	/WTCALC/(	18)	PRINTW STORE WTSCH	I M M	WDECAY WDECAY WDECAY
WDIST1		Fuel system distribution weight pt1	/WTCALC/(	19)	WTSCH	M	WDIST1
WDIST2		Fuel system distribution weight pt2	/WTCALC/(	20)	WTSCH	M	WDIST2
WDOCK		Docking structure weight	/WTCALC/(	21)	PRINTW WTSCH	I M	WDOCK WDOCK
WDPLOY		Deployable aerodynamic device weight	/WTCALC/(	22)	PRINTW WTSCH	I M	WDPLOY WDPLOY
WDRANS		Fuel tank dump and drain weight	/WTCALC/(	23)	WTSCH	M	WDRANS
WDRY		Stage dry weight	/WTCALC/(	24)	PRINTW TAMPER WTSCH	I I M	WDRY WDRY WDRY
WELCAD		Description not input	/WTCALC/(	25)	PRINTW	I	WELCAD
WEMPTY		Stage empty weight	/WTCALC/(	26)	WTSCH	M	WEMPTY
WENGMT		Engine mount weight	/WTCALC/(	27)	WTSCH	M	WENGMT
WENGS		Weight of rocket engines installed	/WTCALC/(	28)	PRINTW WTSCH	I M	WENGS WENGS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
WENGS2		Weight of secondary engines	/WTCALC/(	29)	PRINTW WTSCH	I M	WENGS2 WENGS2
WFAIR		Weight of fairings and shrouds	/WTCALC/(	30)	PRINTW WTSCH	I M	WFAIR WFAIR
WFCONT		Fuel system controls weight	/WTCALC/(	31)	WTSCH	M	WFCONT
WFDCAY		Thrust decay fuel weight	/WTCALC/(	32)	WTSCH	M	WFDCAY
WFROST		Frost and ice weight	/WTCALC/(	33)	PRINTW STORE WTSCH	I M M	WFROST WFROST WFROST
WFU2		Weight of secondary fuel	/WTCALC/(	34)	PRINTW TAMPER WTSCH	I I M	WFU2 WFU2 WFU2
WFUEL		Fuel weight 1. Thrust build-up fuel 2. Not used 3. Main impulse fuel wt. 4. Main impulse fuel reserve 5. Secondary impulse fuel 6. Not used	/WTCALC/(	37)	PRWTSM WTSCH	M M	WFUEL WFUEL
WFUL		Fuel weight	/WTCALC/(	43)	PRINTW WTSCH	I M	WFUL WFUL
WFULOS		Vented fuel	/WTCALC/(	44)	PRINTW STORE WTSCH	I M M	WFULOS WFULOS WFULOS
WFUNCT		Fuel tank weight	/WTCALC/(	45)	WTSCH	M	WFUNCT
WFUOX		Weight of main and secondary propellant	/WTCALC/(	46)	PRINTW TAMPER WTSCH WTVOL	I I M I	WFUOX WFUOX WFUOX WFUOX
WFURES		Fuel reserve	/WTCALC/(	47)	PRINTW STORE TAMPER WTSCH	I M I M	WFURES WFURES WFURES WFURES
WFUSYS		Total fuel system weight	/WTCALC/(	48)	PRINTW WTSCH	I M	WFUSYS WFUSYS
WFUTK		Wt of non-structural fuel tankage	/WTCALC/(	49)	PRINTW WTSCH	I M	WFUTK WFUTK
WFUTK2		Wt of secondary fuel tank and system	/WTCALC/(	50)	PRINTW WTSCH	I M	WFUTK2 WFUTK2
WFUTOT		Total weight of fuel	/WTCALC/(	51)	WTSCH	M	WFUTOT
WFUTRP		Trapped fuel weight	/WTCALC/(	52)	PRINTW STORE WTSCH	I M M	WFUTRP WFUTRP WFUTRP
WGASPR		Weight of gas and pressurant	/WTCALC/(	53)	PRINTW STORE WTSCH	I M M	WGASPR WGASPR WGASPR
WGNAV		Guidance and navigation system wt	/WTCALC/(	54)	PRINTW WTSCH	I M	WGNAV WGNAV
WHORIZ		Horizontal stabilizer wt.	/WTCALC/(	55)	PRINTW PROTHR WTSCH	I I M	WHORZ WHORZ WHORZ
WHYCAD		Hydraulic / pneumatic system wt	/WTCALC/(	56)	PRINTW WTSCH	I M	WHYCAD WHYCAD
WINFUT		Weight of intergral fuel tank	/WTCALC/(	57)	PRINTW PROTHR WTSCH	I I M	WINFUT WINFUT WINFUT
WINOXT		Weight of integral oxidizer tank	/WTCALC/(	58)	PRINTW PROTHR WTSCH	I I M	WINOXT WINOXT WINOXT
WINSTK		Total weight of tank insulation	/WTCALC/(	59)	PRINTW WTSCH	I M	WINSTK WINSTK
WINST		Weight of instrument system	/WTCALC/(	60)	PRINTW WTSCH	I M	WINST WINST
WINSUL		Description not input	/WTCALC/(	61)	PRINTW WTSCH	I M	WINSUL WINSUL

1447

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WJET		Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA PRWTSM STORE TAMPER WTSCH	I M O I M	WJET WJET WJET WJET WJET
WLANCH		Launch gear weight	/WTCALC/(	68)	PRINTW WTSCH	I M	WLANCH WLANCH
WLG		Landing gear and controls weight	/WTCALC/(	69)	PRINTW WTSCH	I M	WLG WLG
WLOSS		In-flight weight loss	/WTCALC/(	70)	PRINTW PRWTSM WTSCH	I I O	WLOSS WLOSS WLOSS
WLRD		Launch and recovery system weight	/WTCALC/(	71)	PRINTW WTSCH	I M	WLRD WLRD
WNACEL		Pylons, nacel,and pod weights	/WTCALC/(	72)	PRINTW WTSCH	I M	WNACEL WNACEL
WODCAY		Oxidizer thrust decay weight	/WTCALC/(	73)	WTSCH	M	WODCAY
WOIL		Service item losses	/WTCALC/(	74)	PRINTW WTSCH	I M	WOIL WOIL
WOILRS		Service item reserves	/WTCALC/(	75)	PRINTW WTSCH	I M	WOILRS WOILRS
WORSUL		Orientation,control, and separation system weight	/WTCALC/(	76)	PRINTW WTSCH	I M	WORSUL WORSUL
WOVERS		Wing loading	/WTCALC/(	77)	PROTHR TAMPER WTSCH	I I M	WOVERS WOVERS WOVERS
WOX		Thrust build-up oxidizer 1. Thrust build-up oxidizer 2. Not used 3. Main impulse oxidizer 4. Main impulse oxidizer reserve 5. Secondary impulse oxidizer 6. Not used	/WTCALC/(	78)	PRWTSM WTSCH	M M	WOX WOX
WOX2		Secondary oxidizer weight	/WTCALC/(	84)	PRINTW TAMPER WTSCH	I I M	WOX2 WOX2 WOX2
WOXID		Main impulse oxidizer weighter	/WTCALC/(	87)	PRINTW WTSCH	I M	WOXID WOXID
WOXLOS		Vented oxidizer	/WTCALC/(	88)	PRINTW STORE WTSCH	I M M	WOXLOS WOXLOS WOXLOS
WOXRES		Oxidizer reserve	/WTCALC/(	89)	PRINTW STORE TAMPER WTSCH	I M I M	WOXRES WOXRES WOXRES WOXRES
WOXSYS		Oxidizer system weight	/WTCALC/(	90)	PRINTW WTSCH	I M	WOXSYS WOXSYS
WOXTK		Non-sturctural tank wt.- oxidizer	/WTCALC/(	91)	PRINTW WTSCH	I M	WOXTK WOXTK
WOXTK2		Secondary system oxidizer tank wt	/WTCALC/(	92)	PRINTW WTSCH	I M	WOXTK2 WOXTK2
WOXTOT		Total weight of oxidizer	/WTCALC/(	93)	WTSCH	M	WOXTOT
WOXTRP		Trapped oxidizer weight	/WTCALC/(	94)	PRINTW STORE WTSCH	I M M	WOXTRP WOXTRP WOXTRP
WP		Total propellant weight	/WTCALC/(	95)	WTSCH	M	WP
WPASS		Weight of passengers	/WTCALC/(	96)	PRINTW WTSCH	I M	WPASS WPASS
WPAYL		Payload weight	/WTCALC/(	97)	PRINTW TAMPER WTSCH	I I M	WPAYL WPAYL WPAYL
WPERS		Crew gear and life support weight	/WTCALC/(	98)	PRINTW WTSCH	I M	WPERS WPERS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WPOWCO		Power conditioning equipment wt.	/WTCALC/(	99)	PRINTW WTSCH	I M	WPOWCO WPOWCO
WPOWER		Description not input	/WTCALC/(	100)	PRINTW WTSCH	I M	WPOWER WPOWER
WPOWFO		Power system propellant wt.	/WTCALC/(	101)	PRINTW STORE WTSCH	I M M	WPOWFO WPOWFO WPOWFO
WPOWRS		Power system propellant reserve	/WTCALC/(	102)	PRINTW STORE WTSCH	I M M	WPOWRS WPOWRS WPOWRS
WPOWTK		Prime power system tank weight	/WTCALC/(	103)	PRINTW WTSCH	I M	WPOWTK WPOWTK
WPPROV		Personnel provisions	/WTCALC/(	104)	PRINTW WTSCH	I M	WPPROV WPPROV
WPREIG		Pre-ignition losses	/WTCALC/(	105)	PRINTW PRWTSM WTSCH	I M M	WPREIG WPREIG WPREIG
WPROP		Total weight- propulsion group	/WTCALC/(	106)	PRINTW WTSCH	I M	WPROP WPROP
WPRSYS		Pressurization system weight	/WTCALC/(	107)	PRINTW WTSCH	I M	WPRSYS WPRSYS
WREFUL		Fuel system refueling system weight	/WTCALC/(	108)	WTSCH	M	WREFUL
WRESID		Weight of residuals	/WTCALC/(	109)	PRINTW WTSCH	I M	WRESID WRESID
WRESRV		Propellant reservs	/WTCALC/(	110)	PRINTW WTSCH	I M	WRESRV WRESRV
WSEAL		Fuel tank seal weight	/WTCALC/(	111)	WTSCH	M	WSEAL
WSECST		Secondary body structure wt	/WTCALC/(	112)	PRINTW PROTHR WTSCH	I I M	WSECST WSECST WSECST
WSORCE		Prime power system weight	/WTCALC/(	113)	PRINTW WTSCH	I M	WSORCE WSORCE
WSRTRP		Trapped oxidizer weight	/WTCALC/(	114)	PRINTW STORE WTSCH	I M M	WSRTRP WSRTRP WSRTRP
WSTAB		Engine gimbal system wt	/WTCALC/(	115)	PRINTW WTSCH	I M	WSTAB WSTAB
WSURF		Aero surface wt	/WTCALC/(	116)	PRINTW WTSCH	I M	WSURF WSURF
WTABC		Net stage weight	/WTCALC/(	117)	WTSCH	O	WTABC
WTHRST		Thrust structure wt	/WTCALC/(	118)	PRINTW WTSCH	I M	WTHRST WTHRST
WTO		Take- off weight	/WTCALC/(	119)	PRINTW WTSCH	I M	WTO WTO
WTPS		Induced environmental protection wt	/WTCALC/(	120)	PRINTW WTSCH	I M	WTPS WTPS
WVERT		Vertical fin weight	/WTCALC/(	121)	PRINTW PROTHR WTSCH	I I M	WVERT WVERT WVERT
WWAIT		Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA PRWTSM STORE TAMPER WTSCH	I I M I M	WWAIT WWAIT WWAIT WWAIT WWAIT
WWET		Operating weight-empty	/WTCALC/(	132)	PRINTW WTSCH	I M	WWET WWET
WWING		Total structural wt. Of wing	/WTCALC/(	133)	PRINTW PROTHR WTSCH	I I M	WWING WWING WWING
WZROFU		Zero fuel weight of vehicle	/WTCALC/(	134)	WTSCH	M	WZROFU

SUBRØUTINE DATAIN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
FRENCH		S	Subroutine to initialize vehicle sizing data	/FRENCH/($	)	DATAIN FRENCH	S E	FRENCH FRENCH
THRUST		S	Subroutine to determine booster, orbiter, and vehicle thrust	/THRUST/($	)	DATAIN THRUST	S E	THRUST THRUST
VEHDF		S	Subroutine to call and initialize synthesis data	/VEHDF /($	)	DATAIN VEHDF	S E	VEHDF VEHDF

```
      PROGRAM DATAIN                                                    CKOUT
C                                                                       DATAIN
C         SUBROUTINE TO READ IN CASE DATA                               DATAIN
C                                                                       DATAIN
      COMMON/GLOBAL/                                                    GLOBAL
     *GR         ,ER        ,OMGZ     ,XLAMRF     ,YMURF     ,LUM       GLOBAL
     *,JJOP(10) ,IFATAL     ,NARC       ,NBRAN     ,NFARC     ,ID(9)    GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG    ,MAXTAB                               GLOBAL
     *,GM        ,PSIRF,IPFLG1,      IPFLG2,IPFLG3,IPFLG4,INEQFL(20)    GLOBAL
     *,ITPSO ,KSOL      ,KGLOBL(8)                                      RETAP
      COMMON/JUMPY/ JUMP,WBIG, WBOD                                     DATAIN
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                    CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,    ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                         CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,     UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,     UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                    CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,    ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                               UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,     UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,      UH
     8TB83,TB84                                                         UH
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCO  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      CALL FRENCH                                                       DATAIN
      CALL VEHDF                                                        DATAIN
      CALL THRUST                                                       DATAIN
      RETURN                                                            DATAIN
      END                                                               DATAIN
```

1452

SUBRØUTINE
FLYBKP

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SCB		M	Working name for input c-array booster scaling coefficients	/ORBINV/(	144)	FLYBKP	M	SCB
						STORE	M	SCB
						SUMOUT	I	SCB
						TAMPER	I	SCB
						THRUST	M	SCB
						VEHDF	I	SCB
						WTVOL	M	SCB
SE		O	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB34		I	Stored booster value of mr(i)	/ORBINV/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34

```
      SUBROUTINE FLYBKP                                                  FLYBKP
C                                                                        FLYBKP
C         SUBROUTINE TO CALCULATE FLYBACK PERFORMANCE PARAMETER          FLYBKP
C                                                                        FLYBKP
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C          PHASE II SIZING PARAMERERS                                    SIZING
     *TZ,       VV(3),     QP(14),    EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),   SQ(37,5),  SE(11),    TLAT,     TLNG,                    SIZING
C         PHASE I SIZING PARAMERERS                                      SIZING
     *WBO,      WLDO,      DWEB,      DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,      TVACB,     NNB,       WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,       DV9,       MUB,       MUO,      VSTG,      WPO              SIZING
     *,JTYP,     BECO,      BSTG,      ORBI,     ITNBW,   ITNOW   ,        SIZING
     * SVOPSQ   , SVOCON   ,IHUNT    ,IOPSTG   ,ISZD(19)                  UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB26(10),TB27(6),TB34(6),        ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                     CKOUT
      COMMON/ORBINY/                                                     ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,     ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27, ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,      UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,       UH
     8TB83,TB84                                                          UH
C                                                                        FLYBKP
C         CRUISE PROPELLANT WEIGHT EQUATION                              FLYBKP
C                                                                        FLYBKP
      FW(A,B,C,D) = WA * (1. - 1./EXP(1.689*A/C*B/D))                    FLYBKP
C                                                                        FLYBKP
      IF( SV(21).NE.0.)  GO TO 11                                        FLYBKP  11
C                                                                        FLYBKP
C         ADJUST FLYBACK RANGE                                           FLYBKP
C                                                                        FLYBKP
      SW(15) = PZ(5) + SQ(10,3)                                          FLYBKP
C                                                                        FLYBKP
C  ****  SW(15) IS THE FLYBACK RANGE  ****                               FLYBKP
C                                                                        FLYBKP
C         DETERMINE FLYBACK PROPELLANT WEIGHT REQUIRED                   FLYBKP
C                                                                        FLYBKP
   11 CONTINUE                                                           FLYBKP
      IFLY = SQ(32,1)                                                    FLYBKP
      GO TO (600,650,700),IFLY                                           FLYBKP  600 650 700
C                                                                        FLYBKP
C         USE BREGUET RANGE EQUATION FOR CRUISE RANGE PARAMETER          FLYBKP
C                                                                        FLYBKP
  600 CONTINUE                                                           FLYBKP
      DMUB = TB34(3) - SQ(19,2)                                          FLYBKP
      IF(SW(3).GT.1.5.AND.ABS(DMUB).GT..001)                             FLYBKP
     1SE(5) = (SW(15) - SQ(19,1))/ DUMB                                  FLYBKP
      SQ(19,1) = SW(15)                                                  FLYBKP
      SQ(19,2) = TB34(3)                                                 FLYBKP
      SQ(19,3) = SW(14)/SW(12)*SW(11)/1.689                              FLYBKP
      SCB(214)=EXP(SW(15)/SQ(19,3))                                      FLYBKP
      SCB(214)= SCB(214) * SV(20)/( SV(20)- SQ(32,4))                    FLYBKP
      SCB(214) = SCB(214) - 1.                                           FLYBKP
      GO TO 800                                                          FLYBKP  800
C                                                                        FLYBKP
C         USE PERCENTAGE WEIGHT METHOD FOR DESCENT AND                   FLYBKP
C         BREGUET RANGE EQUATION FOR CRUISE                              FLYBKP
C                                                                        FLYBKP
  650 CONTINUE                                                           FLYBKP
      DWF = SQ(32,2) * SV(20)                                            CKOUT
      SQ(35,2) = DWF                                                     FLYBKP
      WA = SV(20)                                                        CKOUT
      R2 = SW(15) - SQ(32,5) - SQ(33,1) - SQ(33,2)                       FLYBKP
      DW = FW(R2,SQ(33,4),SQ(34,2),SQ(34,5))                             FLYBKP
      SQ(35,3) = DW                                                      FLYBKP
      WA = WA - DW                                                       FLYBKP
      DWF = DWF + DW                                                     FLYBKP
```

1455

```
      DW = SQ(32,3) * WA                                          FLYBKP
      SQ(35,4) = DW                                               FLYBKP
      GO TO 790                                                   FLYBKP
C                                                                 FLYBKP
C         USE BREGUET EQUATION FOR DESCENT                        FLYBKP
C                                                                 FLYBKP
  700 CONTINUE                                                    FLYBKP
      WA=SV(34)                                                   FLYBKP
      DWF = FW(R1,SQ(33,3),SQ(34,1),SQ(34,4) )                    FLYBKP
      SQ(35,2) = DWF                                              FLYBKP
      WA = SV(20) - DWF                                           CKOUT
      R2 = SW(15) - SQ(32,5) - SQ(33,1) - SQ(33,2)                FLYBKP
      DW = FW(R2,SQ(33,4),SQ(34,2),SQ(34,5))                      FLYBKP
      SQ(35,3) = DW                                               FLYBKP
      DWF = DWF + DW                                              FLYBKP
      WA = WA - DW                                                FLYBKP
      DW = FW(R3,SQ(33,5),SQ(34,3),SQ(35,1) )                     FLYBKP
      SQ(35,4) = DW                                               FLYBKP
C                                                                 FLYBKP
C         CALCULATE CRUISE RANGE PARAMETER                        FLYBKP
C                                                                 FLYBKP
  790 CONTINUE                                                    FLYBKP
      DWF = DWF + DW + SQ(32,4)                                   FLYBKP
      WA = WA - DW - SQ(32,4)                                     FLYBKP
      SCB(214) = DWF/WA                                           FLYBKP
      SQ(19,3) = SQ(34,5)/1.689*SQ(34,2)/SQ(33,4)                 FLYBKP
      SQ(35,5) = R2                                               FLYBKP
  800 CONTINUE                                                    FLYBKP
      RETURN                                                      FLYBKP
      END                                                         FLYBKP
```

SUBRØUTINE FRENCH

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
A		I	Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH FRENCH SETO STORE WTSCH	I M O M I	A ANENGS A ANENGS ANENGS
ANENGS		M	Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH FRENCH SETO STORE WTSCH	I M O M I	A ANENGS A ANENGS ANENGS
FRENCH		E	Subroutine to initialize vehicle sizing data	/FRENCH/($	)	DATAIN FRENCH	S E	FRENCH FRENCH
ID		I	A four word array containing the basic deck, reference run, case and part case numbers in that order.	/GLOBAL/(	21)	BLICO FRENCH GEINP PADSI PRINT SDINP TOPM VEHDF	I I I O I I I I	ID ID ID ID ID ID ID ID
ITPS		O	Thermo protection flag	/CINPUT/(	336)	FRENCH STORE WTSCH	O M M	ITPS ITPS ITPS
JUMP		O	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH PRINTW PRITVA PRWTSM WTSCH WTVOL	O I I M I M	JUMP JUMP JUMP JUMP JUMP JUMP
NWL		O	Wing loading flag	/CINPUT/(	379)	FRENCH STORE WTSCH	O M M	NWL NWL NWL
STORE		S	Subroutine to store vehicle data in internal format and wtvol format	/STORE /($	)	FRENCH STORE	S E	STORE STORE
TRAFLG		O	Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)	FRENCH ITER8 PADSI SIZE SSSP VEHDF	O O I M O O	TRAFLG TRAFLG TRAFLG TRAFLG TRAFLG TRAFLG

1452

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
.UN06.		0	File of all output data	/.UN06./($	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAYO2	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      SUBROUTINE FRENCH                                                  FRENCH
C                                                                        FRENCH
C          SUBROUTINE TO INITIALIZE BOOSTER AND ORBITER                  FRENCH
C          COMMON BLOCKS OF DATA                                         FRENCH
C                                                                        FRENCH
      COMMON/PO/  MPRNT,MPNCH                                            FRENCH
      COMMON/JUMPY/ JUMP,WBIG, WBOO                                      FRENCH
      COMMON/GLOBAL/                                                     GLOBAL
     *GR       ,ER        ,OMGZ      ,XLAMRF     ,YMURF     ,LUM         GLOBAL
     *,JJOP(10) ,IFATAL   ,NARC       ,NBRAN     ,NFARC     ,ID(4)       GLOBAL
     *,KTAB(20),ITAB(20)  ,SIG    ,MAXTAB                                GLOBAL
     *,GM         ,PSIRF,IPFLG1,      IPFLG2,IPFLG3,IPFLG4,INEOFL(20)    GLOBAL
     *,ITPSO ,KSOL     ,KGLOBL(8)                                        RETAP
      REAL KIN                                                           CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                 CINPUT
      REAL NENGS                                                         CINPUT
      COMMON/CINPUT/                                                     CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),    CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,    CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,    CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,    CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,    CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,    CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                           CINPUT
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),        ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                     CKOUT
      COMMON/ORBINX/                                                     ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,     ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,IO22,IO23,IO24,IO25,IO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                          CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,      UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,      UH
     8TO84                                                               UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),        ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                     CKOUT
      COMMON/ORBINY/                                                     ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,     ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,      UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,       UH
     8TB83,TB84                                                          UH
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,    WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,    WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,    WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,    WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFOCAY  ,WFROST  ,    WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,    WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,    WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,    WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRO    ,WNACEL  ,WODCAY  ,    WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,    WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,    WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCO  ,WPOWER  ,WPOWFO  ,    WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,    WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,    WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),   WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,    CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,    CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                         CKOUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,    VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,    VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,    VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,    VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,    VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                           VOLCAL
      DIMENSION A(392)                                                   UH
      EQUIVALENCE (A,ANENGS)                                             UH
      REAL MUB, MUO, ISPB, ISPO, IOVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C           PHASE II SIZING PARAMERERS                                   SIZING
```

1460

```
     *TZ,      VV(3),     QP(14),    EROR,     PZ(5),     VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,     TLNG,                        SIZING
C          PHASE 1 SIZING PARAMERERS                                        SIZING
     *WBO,     WLOO,      OWEB,      OWEO,     TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IOVEL,     ISPO,     ISPB,   SIZING
     *XPL,     TVACB,    NNB,        WEO,      WEB,       WO,       WLO,    SIZING
     *DVO,      DVB,       MUB,       MUO,      VSTG,      WPO              SIZING
     *,JTYP,    BECO ,    BSTG ,     ORBI ,    ITNBW ,  ITNOW  ,            SIZING
     * SVDPSQ  , SVOCON  ,IHUNT    ,IOPSTG  ,ISZD(19)                        UH
      NAMELIST/PRNTX/                                                       UH
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C       ,CBBODY  ,CFUEL   ,       UH
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,       UH
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2           ,FXWOVS  ,       UH
     4ISP     ,ITPS    ,K       ,KIN     ,LF      ,MR      ,NCREW   ,       UH
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,       UH
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,       UH
     7TYTAIL  ,VBODY   ,WGROSS                                              UH
C                                                                           FRENCH
C          READ ORBITER DATA AND STORE IN INTERNAL FORMAT                   FRENCH
C                                                                           FRENCH
      IPC = 31                                                              CKOUT
      JUMP = 0                                                              FRENCH
      CALL SETO                                                             CKOUT
      CALL TABIN(DUM,1,A,392,RUMMY,1,ID,IPC,0,IEOD)                         UH
      IF(IEOD.NE.0) GO TO 1                                                 UH
C                                                                           UH
      MPRNT  = A(205)                                                       UH
      ITPS=A(336)                                                           UH
      NWL= A(379)                                                           UH
      IF(MPRNT.EQ.1) WRITE(6,PRNTX)                                         UH
      IF( ANENGS.EQ. 0. ) ANENGS = 1.                                       CKOUT
       CALL STORE                                                           FRENCH
      CALL ORBSTO                                                           FRENCH
C                                                                           FRENCH
C          READ BOOSTER DATA AND STORE IN INTERNAL FORMAT                   FRENCH
C                                                                           FRENCH
      IPC = 32                                                              CKOUT
      JUMP = 1                                                              FRENCH
      CALL SETO                                                             CKOUT
      CALL TABIN(DUM,1,A,392,RUMMY,1,ID,IPC,0,IEOD)                         UH
      IF(IEOD.NE.0) GO TO 1                                                 UH
C                                                                           UH
      MPRNT  = A(205)                                                       UH
      NWL= A(379)                                                           UH
      ITPS=A(336)                                                           UH
      IF(MPRNT.EQ.1) WRITE(6,PRNTX)                                         UH
      IF( ANENGS.EQ. 0. ) ANENGS = 1.                                       CKOUT
       CALL STORE                                                           FRENCH
      CALL BOOSTO                                                           FRENCH
C                                                                           FRENCH
C          RETURN TO SYNTH PROGRAM                                          FRENCH
C                                                                           FRENCH
      RETURN                                                                FRENCH
    1 WRITE(6,2)                                                            UH
    2 FORMAT(22H NO SIZING DATA INPUT     )                                 UH
      TRAFLG=2.                                                             UH
      CALL SIZERR                                                           UH
      END                                                                   FRENCH
```

SUBRØUTINE
HUNT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
BLOW		I	Idle descent range increment for flyback calculations	/DATA2X/(	70)	HUNT	I	BLOW
EROR		I	Number of booster engines	/SIZING/(	19)	HUNT	I	EROR
						SSSP	I	EROR
						WTVOL	O	EROR
PNOX		I	Solid motor thrust slope	/DATA2X/(	60)	HUNT	I	PNOX
RVAR		I	Solid motor thrust curve intercept	/DATA2X/(	59)	HUNT	I	RVAR
SQRT		F	Square root function	/SQRT /($	)	ANLATM	F	SQRT
						CRASH	F	SQRT
						DCTOE	F	SQRT
						DER3A	F	SQRT
						ENVPRM	F	SQRT
						HUNT	F	SQRT
						MODELA	F	SQRT
						MODELB	F	SQRT
						OPWELL	F	SQRT
						OUT	F	SQRT
						PAT63	F	SQRT
						PAYO2	F	SQRT
						PDBC	F	SQRT
						PDY3A	F	SQRT
						STORE	F	SQRT
						SYMVRT	F	SQRT
						WTSCH	F	SQRT
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      PROGRAM  HUNT                                                    HUNT
C          PROGRAM CONTROLS  HUNTING PROCEEDURE                        HUNT
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                          SIZING
      COMMON /SIZING/                                                  SIZING
C         PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),     QP(14),    EROR,     PZ(5),     VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,     TLNG,                   SIZING
C        PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,      DWEB,      DWEO,     TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IDVEL,     ISPO,     ISPB,  SIZING
     *XPL,     TVACB,     NNB,       WEO,      WEB,       WO,       WLO,   SIZING
     *DVO,     DVB,       MUB,       MUO,      VSTG,      WPO             SIZING
     *,JTYP,   BECO       BSTG       ORBI,     ITNBW ,  ITNOW  ,          SIZING
     * SVOPSQ  , SVDCON   ,IHUNT     ,IOPSTG   ,ISZD(19)                  UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),      ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                   CKOUT
      COMMON/ORBINY/                                                   ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,   ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                              UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,    UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,     UH
     8TB83,TB84                                                        UH
      REAL NXFOB                                                       DATA2X
      REAL      IVACO,     IVACB,     ISLO,      ISLB,      IDVELO     DATA2X
      COMMON/DATA2X/                                                   DATA2X
     *ALD, FBPAR, IDVELO, ISLB, ISLO, IVACB, IVACO, PERISP, QMXX,      DATA2X
     1 QMXS,SFC,SLVOUT,COPIES,SYNIT,TFCTRB,TFCTRO,TOLMU,TOLTW,         DATA2X
     2 TRATIO,TWLO,TWLOI,WTOUT,FIRE,BOOTW,VCRUSE                       DATA2X
     3 ,NXFOB,PRNTX,FSEC,CLVG,DRNG                                     DATA2X
     4 ,SOLID,AS,BS,SISP,SINERT,SAE,TSBO,FLYBCK                        DATA2X
     5 ,WPOREQ,WOREQ,GWREQ                                             DATA2X
     6 ,FBFUEL,CA,CB,WFLYX,RT,R1,R3,SFC1,SFC2,SFC3,ALD1,ALD2,ALD3,VFLY1, DATA2X
     7 VFLY2,VFLY3                                                     DATA2X
     *,TWOX(2),PNDX(10),BLOW(10),BUPP(10),STEP(11),PAYX                POW
      EQUIVALENCE (RVAR,TWOX(2))                                       POW
       DIMENSION  A(1),XVAR(10),PVAR(10),STAP(11),EROR(10)             HUNT
      EQUIVALENCE (A, TB1)                                             HUNT
      COMMON/POWEL/ IZ(45)                                             HUNT
      EQUIVALENCE (IZ(31),CON)        ,(IZ(45),EN)                     HUNT
      INTEGER EN ,CON                                                  HUNT
C          FUNCTIONS                                                   HUNT
      BOXIN(XX,II) =  ASIN(SQRT((XX - BLOW(II))/(BUPP(II)-BLOW(II))))  HUNT
      BOX(XX,II)   =    BLOW(II) + (BUPP(II) - BLOW(II))* SIN(XX)**2   HUNT
      NVAR=RVAR                                                        HUNT
C     I    PUT    A VALUES  INTO  XVAR  ARRAY                          HUNT
      DO 10 I=1,NVAR                                                   HUNT
      INDX = PNDX(I)                                                   HUNT
      XVAR(I)=A(INDX)                                                  HUNT
      PVAR(I) = BOXIN(XVAR(I),I)                                       HUNT
      STAP(I)=BOXIN(STEP(I),I)                                         HUNT
   10 CONTINUE                                                         HUNT
      INDPA =ABS(PAYX)                                                 HUNT
      SI=1.                                                            HUNT
      IF(PAYX.GT.0.) SI=-1.                                            HUNT
      PAY =  A(INDPA)*SI                                               HUNT
      CALL   POWELL(PVAR,NVAR,STAP,EROR,PAY)                           HUNT
C         TEST  FOR  CONVERGANCE                                       HUNT
      DO 20 I=1,NVAR                                                   HUNT
      INDX = PNDX(I)                                                   HUNT
      A(INDX) = BOX(PVAR(I),I)                                         HUNT
      STEP(I)=BOX(STAP(I),I)                                           HUNT
   20 CONTINUE                                                         HUNT
      IF(CON.NE.-1) RETURN                                             HUNT
      PAYO=A(INDPA)                                                    HUNT
      WRITE(6,103) PAYO                                                HUNT
  103 FORMAT(35H0 POWELL HAS CONVERGED TO A PAYOFF=E20.8)              HUNT
      RETURN                                                           HUNT
      END                                                              HUNT
```

SUBRØUTINE
ITER8

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ITER8		E	Subroutine to evaluate synthesis convergance and call for flyback range calculations and summary page output	/ITER8 /($	)	ITER8	E	ITER8
RANGE		S	Subroutine to evaluate booster flyback range based on flybck	/RANGE /($	)	ITER8	S	RANGE
						RANGE	E	RANGE
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB34		O	Stored booster value of mr(I)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TOLER		W	Orbiter mass ratio toleranceachieved	/ITER8 /(*	)	ITER8	W	TOLER
TO34		M	Stored orbiter value of mr(I)	/ORBINX/(	53)	ITER8	M	TO34
						SSSP	O	TO34
						STORE	M	TO34
						TAMPER	I	TO34
						VEHDF	O	TO34
						WTVOL	M	TO34
TRAFLG		O	Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)	FRENCH	O	TRAFLG
						ITER8	O	TRAFLG
						PADSI	I	TRAFLG
						SIZE	M	TRAFLG
						SSSP	O	TRAFLG
						VEHDF	O	TRAFLG

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
.UN06.		0	File of all output data	/.UN06./(S	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAY02	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      PROGRAM ITER8                                                    CKOUT
C                                                                      ITER8
C         SUBROUTINE TO CHECK SYNTHESIS LOOP FOR CONVERGENCE           ITER8
C                                                                      ITER8
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                          SIZING
      COMMON /SIZING/                                                  SIZING
C          PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,      VV(3),     OP(14),    EROR,     PZ(5),     VO,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,     TLNG,                   SIZING
C         PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,     WLOO,      DWEB,      DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,   NNB,         WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,       MUO,      VSTG,      WPO             SIZING
     *,JTYP,    BECO       BSTG       ORBI,    ITNBW  , ITNOW  ,           SIZING
     * SVDPSQ  , SVDCON  ,IHUNT    ,IOPSTG   ,ISZD(19)                     UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),      ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                   CKOUT
      COMMON/ORBINX/                                                   ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,   ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                         CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,    UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,    UH
     8TO84                                                             UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),      ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                   CKOUT
      COMMON/ORBINY/                                                   ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,   ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                              UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,    UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,     UH
     8TB83,TB84                                                        UH
C                                                                      ITER8
C     SW(2) = 2 FOR CONVERGED RUN                                      ITER8
C     SW(2)=1. FOR LAST PASS                                           ITER8
C     SW(2) =.5 FOR ITERATION                                          ITER8
C                                                                      ITER8
C         CALCULATE IMPULSE MASS RATIOS                                ITER8
C                                                                      ITER8
      CALL SIZEMR                                                      ITER8
C                                                                      ITER8
C         CALC ERROR IN MASS RATIO AND PRINT                           ITER8
C                                                                      ITER8
      TOLER = ABS( TO34(3) - SV(6) ) / SV(6)                           ITER8
      WRITE (6,1005) SV(6),TO34(3), SW(3),SW(5),TOLER                  ITER8
C                                                                      ITER8
C         CHECK FOR SUCCESSFUL CONVERGENCE                             ITER8
C                                                                      ITER8
      IF(TOLER.LT.SW(5))  GO TO 1                                      CKOUT
      GO TO 11                                                         ITER8
C                                                                      ITER8
    1 SW(2) = 2.                                                       CKOUT
      TRAFLG = 1.0                                                     CKOUT
      GO TO 11                                                         CKOUT
C         SET ORBITER MASS RATIO FOR WTVOL ROUTINES                    ITER8
C                                                                      ITER8
C         CALL FLYBACK RANGE                                           ITER8
C                                                                      ITER8
   11 TO34(3) = SV(6)                                                  ITER8
      TB34(3) = SV(28)                                                 UH
      CALL RANGE                                                       ITER8
      CALL FLYBKP                                                      ITER8
      IF( SW(2).GE. 1.) RETURN                                         ITER8
C                                                                      ITER8
C         IF MASS RATIO ERROR TOO LARGE PRINT                          ITER8
C                                                                      ITER8
```

```
   500 IF( TOLER .GT. SW(5) ) WRITE(6,1000) SV(6),T034(3),SW(5),TOLER      ITER8
C                                                                          ITER8
C          IF ITERATION COUNTER EXCEEDED PRINT                             ITER8
C                                                                          ITER8
      IF(SW(3).GT.SW(4)) WRITE(6,1021) SW(3),SW(5),TOLER                   ITER8
C                                                                          ITER8
C          RETURN TO SYNTH PROGRAM                                         ITER8
      RETURN                                                               ITER8
 1000 FORMAT (/26H FAILED TO CONVERGE - MU = 2E16.8,                       ITER8
     1 5X,13H INPUT TOL = F8.5,5X,7H TOL = F9.6)                           ITER8
 1005 FORMAT(/5X,10H TOP MU = F10.6,5X,11HWTVOL MU = F10.6,F4.0,           CKOUT
     1 5X,13H INPUT TOL = F8.5,5X,7H TOL = F9.6)                           ITER8
 1020 FORMAT (//31H  ITERATION COUNTER EXCEEDED = F4.0//)                  ITER8
 1021 FORMAT (//31H  ITERATION COUNTER EXCEEDED = F4.0,                    ITER8
     1 5X,13H INPUT TOL = F8.5,5X,7H TOL = F9.6)                           ITER8
      END                                                                  ITER8
```

1467

SUBRØUTINE ØPWELL

1475

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SQRT		F	Square root function	/SQRT	/($)	ANLATM	F	SQRT
						CRASH	F	SQRT
						DCTDE	F	SQRT
						DER3A	F	SQRT
						ENVPRM	F	SQRT
						HUNT	F	SQRT
						MODELA	F	SQRT
						MODELB	F	SQRT
						OPWELL	F	SQRT
						OUT	F	SQRT
						PAT63	F	SQRT
						PAYD2	F	SQRT
						PDBC	F	SQRT
						PDY3A	F	SQRT
						STORE	F	SQRT
						SYMVRT	F	SQRT
						WTSCH	F	SQRT

```
      SUBROUTINE OPWELL(PV,N,STEP,EPP,PAY)                              OPWELL
      COMMON /OPWEL/                                                    OPWELL
     *DEL (11) ,II      ,COUNT    ,XT(3)    ,AT (3)    ,PVS(10)  ,     JULY28
     *PASAV    ,CON     ,LOU    ,EN    ,PPN(10)  ,                      JULY28
     *SC    ,MOUNT    ,GG(11,10)                                        JULY28
     *  , PAO, NT  , NC  , DD  , D(7) , V(7)                            JULY28
      COMMON/SIZING/DUMMY(320),ELIM                                     JULY28
      DIMENSION PV(1),STEP(1)                                           JULY28
     *,EPP(1)                                                           JULY28
      INTEGER  COUNT,CON,EN                                             JULY28
      INTEGER ELIM                                                      JULY28
C                                                                       JULY28
C            BEGIN LOGIC FOR SUBROUTINE POWELL                          JULY28
C                                                                       JULY28
      IF(EN.GE.1) GO TO 19                                              JULY28
C                                                                       JULY28
C            FIRST PASS LOGIC                                           JULY28
C                                                                       JULY28
      NC=0                                                              JULY28
      NT = 0                                                            JULY28
      PAO =PAY                                                          JULY28
      SC=1.                                                             JULY28
      CON = -1                                                          JULY28
      LOU=0                                                             JULY28
      COUNT =0                                                          JULY28
      MOUNT =0                                                          JULY28
      II=1                                                              JULY28
      EN=1                                                              JULY28
      DO 5  I=1,3                                                       JULY28
      XT(I) = 0.                                                        JULY28
    5 AT(I) = 0.                                                        JULY28
      DO 6  I=1,11                                                      JULY28
    6 DEL(I) = 0.                                                       JULY28
      DO 10  I=1,N                                                      JULY28
      PVS(I) =PV(I)                                                     JULY28
   10 PPN(I) = PV(I)                                                    JULY28
      NP1 =N+1                                                          JULY28
      DO 17 I=1,NP1                                                     JULY28
      DO 17  J =1,N                                                     JULY28
      IF(I.EQ.J)  GO TO 15                                              JULY28
      GG(I,J)= 0.                                                       JULY28
      GO TO 17                                                          JULY28
   15 GG(I,J)= 1.                                                       JULY28
   17 CONTINUE                                                          JULY28
C                                                                       JULY28
C            CHECK VALUE OF MOUNT.                                      JULY28
C                          1.) IF MOUNT EQUAL ZERO STORE INFORMATION    JULY28
C                              REQUIRED FOR ACCELERATED LINEAR SEARCH.  JULY28
C                          2 ) IF MOUNT EQUAL ONE CONTINUE WITH QUADRATIC JULY28
C                              SEARCH.                                  JULY28
C                                                                       JULY28
   19 IF(MOUNT.EQ.1) GO TO 100                                          JULY28
   20 COUNT = COUNT +1                                                  JULY28
      XT(1) = XT(2)                                                     JULY28
      XT(2) = XT(3)                                                     JULY28
      XT(3) = DEL(II)                                                   JULY28
      AT(1) = AT(2)                                                     JULY28
      AT(2) = AT(3)                                                     JULY28
      AT(3) = PAY                                                       JULY28
      IF(COUNT.GE.2) GO TO 70                                           JULY28
C                                                                       JULY28
C            SAVE PAYOFF TO CHECK IF ACCELERATED LINEAR SEARCH          JULY28
C            SHOULD BE ENDED.                                           JULY28
C                                                                       JULY28
   52 PASAV = PAY                                                       JULY28
C                                                                       JULY28
C            COMPUTE STEP SIZE FOR USE IN ACCELERATED LINEAR SEARCH,    JULY28
C            AND COMPUTE NEW VALUE OF PV.                               JULY28
C                                                                       JULY28
      VAL = 0.                                                          JULY28
      DO 53  I=1,N                                                      JULY28
   53 VAL = VAL + (GG(II,I)/STEP(I))**2                                 JULY28
      DEL(II) =1./SQRT(VAL)                                             JULY28
```

```
      DEL(II) = 2.**COUNT*DEL(II)/(2.*EN)                       JULY28
      DEL(II) = SC * DEL(II)                                    JULY28
      DO 55 I =1,N                                              JULY28
   55 PV(I)= PPN(I) + DEL(II)*GG(II,I)                          JULY28
      RETURN                                                    JULY28
C                                                               JULY28
C             COMPARE VALUES FROM ACCELERATED LINEAR SEARCH.    JULY28
C                1.) IF FIRST THREE POINTS FORM A CONCAVE UPWARD JULY28
C                    SET, GO TO QUADRATIC SEARCH.               JULY28
C                2.) IF PAY IS LESS THAN PASAV, SET PASAV EQUAL JULY28
C                    TO PAY, COMPUTE NEW STEP SIZE AND NEW VALUE JULY28
C                    OF PV.                                     JULY28
C                3.) IF PAY IS GREATER THAN OR EQUAL TO PASAV,  JULY28
C                        A.)  IF COUNT IS EQUAL TO TWO, SET     JULY28
C                             PASAV EQUAL TO PAY, REVERSE THE   JULY28
C                             DIRECTION OF NEW STEP,COMPUTE     JULY28
C                             NEW PV.                           JULY28
C                        B.)  IF COUNT NOT EQUAL TO TWO BEGIN   JULY28
C                             QUADRATIC SEARCH.                 JULY28
C                                                               JULY28
   70 IF(PAY.GT.AT(1).AND.SC.EQ.-.5.AND.COUNT.EQ.3) GO TO 75    JULY28  75
      IF(PAY.LT.PASAV) GO TO 52                                 JULY28  52
      IF(COUNT.EQ.2) GO TO 90                                   JULY28  90
C                                                               JULY28
C             BEGIN QUADRATIC SEARCH.                           JULY28
C                                                               JULY28
   75 NC=NC+1                                                   JULY28
      AA = (XT(2)-XT(3))*AT(1)                                  JULY28
      BB =(XT(3) - XT(1))*AT(2)                                 JULY28
      CC =(XT(1) - XT(2))*AT(3)                                 JULY28
      AS =(XT(2)**2 -XT(3)**2 )*AT(1)                           JULY28
      BS =(XT(3)**2 -XT(1)**2 )*AT(2)                           JULY28
      CS =(XT(1)**2 -XT(2)**2 )*AT(3)                           JULY28
      DD= (AS+BS+CS)/(AA+BB+CC)                                 JULY28
      DD = .5*DD                                                JULY28
      D(NC) =DD                                                 JULY28
      PASAV = PAY                                               JULY28
      DO 85 I=1,N                                               JULY28
   85 PV(I) = PPN(I) +DD*GG(II,I)                               JULY28
      MOUNT = 1                                                 JULY28
      RETURN                                                    JULY28
C                                                               JULY28
C             REVERSE DIRETION OF STEP.                         JULY28
C                                                               JULY28
   90 SC = -.5                                                  JULY28
      GO TO 52                                                  JULY28  52
C                                                               JULY28
C             CHECK VALUE OF LOU.                               JULY28
C                1.) IF LOU EQUAL TO ONE, CHECK CONVERGENCE OF  JULY28
C                    QUADRATIC SEARCH                           JULY28
C                2.) IF LOU EQUAL TO ZERO, THROW AWAY WORST     JULY28
C                    VALUE OF ACCELERATED LINEAR SEARCH AND     JULY28
C                    REPLACE BY RESULT OF QUADRATIC SEARCH      JULY28
C                                                               JULY28
  100 V(NC)= PAY                                                JULY28
      IF(LOU.EQ.1) GO TO 150                                    JULY28  150
  105 PMAX = AMAX1(AT(1), AT(2), AT(3) )                        JULY28
      DO 120 I=1,3                                              JULY28
      IF(PMAX.EQ. AT(I)) GO TO 121                              JULY28  121
  120 CONTINUE                                                  JULY28
  121 JM = I                                                    JULY28
      PMIN = AMIN1(AT(1),AT(2),AT(3))                           JULY28
      DO 130  I =1,3                                            JULY28
      IF(PMIN.EQ.AT(I))  GO TO 131                              JULY28  131
  130 CONTINUE                                                  JULY28
  131 JS = I                                                    JULY28
      JL = MOD(JM+JS,4)                                         JULY28
      IF(JL.EQ.0) JL =2                                         JULY28
      FM1 = AMAX1(XT(1),XT(2),XT(3))                            JULY28
      FM2 = AMIN1(XT(1),XT(2),XT(3))                            JULY28
      IF(DD.GT.FM1.OR.DD.LT.FM2) GO TO 141                      JULY28  141
      IF(XT(JM).EQ.FM1.AND.XT(JL).EQ.FM2) GO TO 137             JULY28  137
```

```
      IF(XT(JM).EQ.FM2.AND.XT(JL).EQ.FM1) GO TO 135                     JULY28
      GO TO 141                                                         JULY28
  135 IF(DD.GT.XT(JS)) GO TO 141                                        JULY28
      GO TO 145                                                         JULY28
  137 IF(DD.GT.XT(JS)) GO TO 145                                        JULY28
  141 XT(JM) =DD                                                        JULY28
      AT(JM) =PAY                                                       JULY28
      GO TO 149                                                         JULY28
  145 XT(JL) =DD                                                        JULY28
      AT(JL)= PAY                                                       JULY28
  149 LOU =1                                                            JULY28
      GO TO 75                                                          JULY28
  150 IF(NC.GE.7) GO TO 152                                             JULY28
      IF(PAY.GT.PASAV) GO TO 105                                        JULY28
      TT=.2                                                             JULY28
      IF(N.EQ.1) TT=.05                                                 JULY28
      IF(PASAV-PAY.GT.TT*PASAV) GO TO 105                               JULY28
      IF(NT.EQ.2) GO TO 152                                             JULY28
      NT = NT+1                                                         JULY28
      IF(PAY.GT.PAO) GO TO 105                                          JULY28
C                                                                       JULY28
C           IF N EQUAL TO ONE, SET CON EQUAL TO ONE AND RETURN          JULY28
C                                                                       JULY28
  152 IM=1                                                              JULY28
      VMIN=V(1)                                                         JULY28
      DO 153 I=2,NC                                                     JULY28
      IF(VMIN.GT.V(I)) IM=I                                             JULY28
  153 VMIN = V(IM)                                                      JULY28
      DD = D(IM)                                                        JULY28
      DO 154  I=1,N                                                     JULY28
  154 PV(I) = PPN(I)+DD* GG(II,I)                                       JULY28
      PAY=V(IM)                                                         JULY28
      NT=0                                                              JULY28
      NC=0                                                              JULY28
      IF(N.EQ.1) GOTO 252                                               JULY28
      IF(II.EQ.N+1) GO TO 200                                           JULY28
C                                                                       JULY28
C           RESET FOR NEW MINOR OR COMBINED SEARCH.                     JULY28
C                                                                       JULY28
      DO 155  I=1,N                                                     JULY28
  155 PPN(I) = PPN(I) + DD*GG(II,I)                                     JULY28
      NT = 0                                                            JULY28
      SC=1.                                                             JULY28
      COUNT =0                                                          JULY28
      MOUNT = 0                                                         JULY28
      LOU = 0                                                           JULY28
      NP1=N+1                                                           JULY28
      DO 160 I=1,NP1                                                    JULY28
  160 DEL(I) = 0.                                                       JULY28
      II=II+1                                                           JULY28
      IF(II.LE.N) GO TO 20                                              JULY28
C                                                                       JULY28
C           COMPUTE COMBINED DIRETION.                                  JULY28
C                                                                       JULY28
      GVAL=0                                                            JULY28
      DO 175 I = 1,N                                                    JULY28
  175 GVAL = GVAL+(PPN(I)-PVS(I))**2                                    JULY28
      GVAL = SQRT(GVAL)                                                 JULY28
      DO 180 I = 1,N                                                    JULY28
  180 GG(II,I) = (PPN(I)-PVS(I))/GVAL                                   JULY28
      GO TO 20                                                          JULY28
C                                                                       JULY28
C           CHECK CONVERGENCE  OF PROBLEM.                              JULY28
C                                                                       JULY28
  200 DO 250 I =1,N                                                     JULY28
  250 IF(ABS(PVS(I)-PV(I)).GT.EPP(I)) GO TO 255                         JULY28
  252 CON = 1                                                           JULY28
      RETURN                                                            JULY28
C                                                                       JULY28
C           INCREMENT EN.                                               JULY28
C                   1.) IF EN IS LESS THAN OR EQUAL TO ELIM RESET       JULY28
```

```
      C                              FOR NEW MAJOR                          JULY28
      C                         2.) IF NOT,SET CON EQUAL TO ZERO AND RETURN.  JULY28
      C                                                                     JULY28
        255 EN=EN+1                                                         JULY28
            IF(EN.LE.ELIM ) GO TO 260                                       JULY28  260
            CON = 0                                                         JULY28
            RETURN                                                          JULY28
        260 DO 261 I = 1,N                                                  JULY28
            PVS(I) =PV(I)                                                   JULY28
        261 PPN(I) = PV(I)                                                  JULY28
            NT = 0                                                          JULY28
            PAO =PAY                                                        JULY28
            II =1                                                           JULY28
            SC = 1.                                                         JULY28
            COUNT=0                                                         JULY28
            MOUNT = 0                                                       JULY28
            LOU   = 0                                                       JULY28
            NP1=N+1                                                         JULY28
            DO 265  I=1,NP1                                                 JULY28
        265 DEL(I) = 0.                                                     JULY28
            DO 270 I = 1,N                                                  JULY28
            DO 270  J =1,N                                                  JULY28
        270 GG(I,J) = GG(I+1,J)                                             JULY28
            GO TO 20                                                        JULY28  20
            END                                                             JULY28
```

1475

SUBRØUTINE
PRINTV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
KILO		W	Pounds to kilogram conversion	/PRINTV/(*	)	PRINTV	W	KILO
KVBODY		W	Same as vbody in mks units	/PRINTV/(*	)	PRINTV	W	KVBODY
KVCARG		W	Same as vcargo in mks units	/PRINTV/(*	)	PRINTV	W	KVCARG
KVCREW		W	Same as vcrew in mks units	/PRINTV/(*	)	PRINTV	W	KVCREW
KVFUTK		W	Same as vfutk in mks units	/PRINTV/(*	)	PRINTV	W	KVFUTK
KVFUT2		W	Same as vfutk2 in mks units	/PRINTV/(*	)	PRINTV	W	KVFUT2
KVINST		W	Same as vinstk in mks units	/PRINTV/(*	)	PRINTV	W	KVINST
KVLGBA		W	Same as vlgbay in mks units	/PRINTV/(*	)	PRINTV	W	KVLGBA
KVOTHE		W	Same as vother in mks units	/PRINTV/(*	)	PRINTV	W	KVOTHE
KVOXTK		W	Same as voxtk in mks units	/PRINTV/(*	)	PRINTV	W	KVOXTK
KVOXT2		W	Same as voxtk2 in mks units	/PRINTV/(*	)	PRINTV	W	KVOXT2
KVPROP		W	Same as vprop in mks units	/PRINTV/(*	)	PRINTV	W	KVPROP
KVSTRU		W	Same as vstruc in mks units	/PRINTV/(*	)	PRINTV	W	KVSTRU
PRINTV		E	Subroutine to print vehicle volume data	/PRINTV/($	)	PRINTV WTVOL	E S	PRINTV PRINTV
PROTHR		S	Subroutine to print misc data	/PROTHR/($	)	PRINTV PROTHR	S E	PROTHR PROTHR
VBODY		M	Total body volume	/CINPUT/(	391)	PRINTV SOLVE STORE TAMPER WTSCH WTVOL	M M M I M I	VBODY VBODY VBODY VBODY VBODY VBODY
VCARGO		I	Volume of cargo bay	/VOLCAL/(	27)	PRINTV TAMPER WTSCH	I I M	VCARGO VCARGO VCARGO
VCREW		I	Volume of crew compartment	/VOLCAL/(	28)	PRINTV WTSCH	I M	VCREW VCREW
VFUTK		I	Total volume of fuel tank	/VOLCAL/(	29)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK VFUTK VFUTK VFUTK
VFUTK2		I	Total volume of secondary fuel tank	/VOLCAL/(	30)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK2 VFUTK2 VFUTK2 VFUTK2
VINSTK		I	Total tank insulation volume	/VOLCAL/(	31)	PRINTV WTSCH	I M	VINSTK VINSTK
VLGBAY		I	Volume of recovery system bay	/VOLCAL/(	32)	PRINTV WTSCH	I M	VLGBAY VLGBAY
VOXTK		I	Total volume of oxidizer tank	/VOLCAL/(	34)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK VOXTK VOXTK VOXTK
VOXTK2		I	Total volume of secondary oxidizer tank	/VOLCAL/(	35)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK2 VOXTK2 VOXTK2 VOXTK2
VPROP		I	Volume of propulsion bay	/VOLCAL/(	36)	PRINTV WTSCH	I M	VPROP VPROP
VSTRUC		I	Volume of basic structure	/VOLCAL/(	37)	PRINTV WTSCH	I M	VSTRUC VSTRUC

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
.UN06.		0	File of all output data	/.UN06./(S	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAY02	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      SUBROUTINE PRINTV                                                  PRIN
C ***  PRINTV  ***                                                       PRIN
C                                                                        PRIN
C    PRINTV - PRINT  VOLUMES                                             PRIN
C                                                                        PRIN
      REAL KILO,KVSTRU,KVCARG,KVLGBA,KVPROP,KVFUTK,KVOXTK,KVINST,KVOTHE  PRIN
     *,KVBODY,KVFUT2,KVOXT2,KVCREW                                       PRIN
      REAL KIN                                                           CINP
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                 CINP
      REAL NENGS                                                         CINP
      COMMON/CINPUT/                                                     CINP
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),    CINP
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,    CINP
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,    CINP
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,    CINP
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,    CINP
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,    CINP
     7TYTAIL  ,VBODY   ,WGROSS                                           CINP
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,    VOLC
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,    VOLC
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,    VOLC
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,    VOLC
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,    VOLC
     6VOXTK2  ,VPROP   ,VSTRUC                                           VOLC
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,    WTCA
     1WACS    ,WACSFD  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,    WTCA
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,    WTCA
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,    WTCA
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,    WTCA
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,    WTCA
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,    WTCA
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,    WTCA
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,    WTCA
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,    WTCA
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,    WTCA
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCO  ,WPOWER  ,WPOWFO  ,    WTCA
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,    WTCA
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,    WTCA
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),   WTCA
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,    CKOU
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,    CKOU
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                         CKOU
    9 KILO = .0283                                                       PRIN
      KVSTRU =KILO *VSTRUC                                               PRIN
      KVCREW=KILO *VCREW                                                 PRIN
      KVOXT2=KILO*VOXTK2                                                 PRIN
      KVFUT2=KILO*VFUTK2                                                 PRIN
      KVCARG =KILO *VCARGO                                               PRIN
      KVLGBA =KILO *VLGBAY                                               PRIN
      KVPROP =KILO *VPROP                                                PRIN
      KVFUTK =KILO *VFUTK                                                PRIN
      KVOXTK =KILO *VOXTK                                                PRIN
      KVINST = VINSTK*KILO                                               PRIN
      KVOTHE =KILO *VOTHER                                               PRIN
      KVBODY =KILO *VBODY                                                PRIN
      WRITE (6,8)                                                        PRIN
      WRITE(6,10)                                                        PRIN
      WRITE(6,100) VSTRUC,KVSTRU                                         PRIN
      WRITE(6,200) VCREW,KVCREW                                          PRIN
      WRITE (6,300) VCARGO,KVCARG                                        PRIN
      WRITE(6,400) VLGBAY,KVLGBA                                         PRIN
      WRITE(6,500) VPROP,KVPROP                                          PRIN
      WRITE(6,700) VFUTK,KVFUTK                                          PRIN
      WRITE(6,800) VOXTK,KVOXTK                                          PRIN
      WRITE(6,900) VINSTK,KVINST                                         PRIN
      WRITE(6,1400) VFUTK2,KVFUT2                                        PRIN
      WRITE(6,1500) VOXTK2,KVOXT2                                        PRIN
 1110 FORMAT(1H0,40HTOTAL ENTRY VEHICLE VOLUME             F9.1,F10.1) PRIN
  950 WRITE (6,1000)VOTHER,KVOTHE                                        PRIN
      WRITE(6,1110) VBODY,KVBODY                                         PRIN
      IF(VBODY.EQ.0.) VBODY = 1.                                         PRIN
      CALL PROTHR                                                        PRIN
      RETURN                                                             PRIN
    8 FORMAT (1H1)                                                       PRIN
   10 FORMAT(    8H VOLUMES  ,35X,10H  CU FT     ,10H  CU M        /   ) PRIN
```

```
   100 FORMAT(1H0,40HBODY STRUCTURE                              2F10.2)     PRINT
   200 FORMAT(1H ,40HCREW AND PASSENGER COMPARTMENTS             2F10.2)     PRINT
   300 FORMAT(1H ,40HCARGO COMPARTMENT                           2F10.2)     PRINT
   400 FORMAT(1H ,40HLANDING GEAR BAYS                           2F10.2)     PRINT
   500 FORMAT(1H ,40HPROPULSION BAY WITHIN BODY                  2F10.2)     PRINT
 700   FORMAT(1H ,40HFUEL CONTAINERS                                 2F10.2) PRINT
 800   FORMAT(1H ,40HOXIDIZER CONTAINERS                             2F10.2) PRINT
 900   FORMAT(1H ,40HPROPELLANT INSULATION                           2F10.2) PRINT
  1000 FORMAT(1H ,40HOTHER BODY VOLUME                           2F10.2)     PRINT
  1400 FORMAT(1H ,40HSECONDARY FUEL CONTAINERS                       2F10.2) PRINT
  1500 FORMAT(1H ,40HSECONDARY OXIDIZER CONTAINERS                   2F10.2) PRINT
       END                                                                   PRINT
```

1480

SUBRØUTINE PRINTW

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
C		I	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
						PRITEQ	I	C
						PRITVA	I	C
						STORE	M	C
						WTSCH	I	C
						WTVOL	O	C
FROST		W	Wt of ice and frost (kg)	/PRINTW/(*	)	PRINTW	W	FROST
JUMP		I	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH	O	JUMP
						PRINTW	I	JUMP
						PRITVA	I	JUMP
						PRWTSM	M	JUMP
						WTSCH	I	JUMP
						WTVOL	M	JUMP
KILO		W	Pounds to kilogram conversion	/PRINTW/(*	)	PRINTW	W	KILO
KOXTK2		W	Same as woxtk2 in mks units	/PRINTW/(*	)	PRINTW	W	KOXTK2
KWABFT		W	Same as wabftk in mks units	/PRINTW/(*	)	PRINTW	W	KWABFT
KWABPR		W	Same as wabpr in mks units	/PRINTW/(*	)	PRINTW	W	KWABPR
KWACRE		W	Same as wacres in mks units	/PRINTW/(*	)	PRINTW	W	KWACRE
KWACS		W	Same as wacs in mks units	/PRINTW/(*	)	PRINTW	W	KWACS
KWACSF		W	Smae as wacsfo in mks units	/PRINTW/(*	)	PRINTW	W	KWACSF
KWACST		W	Same as wacstk in mks units	/PRINTW/(*	)	PRINTW	W	KWACST
KWAERO		W	Same as waero in mks units	/PRINTW/(*	)	PRINTW	W	KWAERO
KWAUXT		W	Same as wauxt in mks units	/PRINTW/(*	)	PRINTW	W	KWAUXT
KWBASI		W	Same as wbasic in mks units	/PRINTW/(*	)	PRINTW	W	KWBASI
KWBODY		W	Same as wbody in mks units	/PRINTW/(*	)	PRINTW	W	KWBODY
KWCARG		W	Same as wcargo in mks units	/PRINTW/(*	)	PRINTW	W	KWCARG
KWCOMM		W	Same as wcomm in mks units	/PRINTW/(*	)	PRINTW	W	KWCOMM
KWCONT		W	Same as wcont in mks units	/PRINTW/(*	)	PRINTW	W	KWCONT
KWCOVE		W	Same as wcover in mks units	/PRINTW/(*	)	PRINTW	W	KWCOVE
KWDECA		W	Same as wdecay in mks units	/PRINTW/(*	)	PRINTW	W	KWDECA
KWDOCK		W	Same as wdock in mks units	/PRINTW/(*	)	PRINTW	W	KWDOCK
KWDPLO		W	Same as wdploy in mks units	/PRINTW/(*	)	PRINTW	W	KWDPLO
KWDRY		W	Same as wdry in mks units	/PRINTW/(*	)	PRINTW	W	KWDRY
KWELCA		O	Same as welcad in mks units	/PRINTW/(*	)	PRINTW	O	KWELCA
KWENGS		W	Same as wengs in mks units	/PRINTW/(*	)	PRINTW	W	KWENGS
KWFAIR		W	Same as wfair in mks units	/PRINTW/(*	)	PRINTW	W	KWFAIR
KWFUL		W	Same as wful in mks units	/PRINTW/(*	)	PRINTW	W	KWFUL
KWFULO		W	Same as wfulos in mks units	/PRINTW/(*	)	PRINTW	W	KWFULO
KWFUOX		W	Same as wfuox in mks units	/PRINTW/(*	)	PRINTW	W	KWFUOX
KWFURE		W	Same as wfures in mks units	/PRINTW/(*	)	PRINTW	W	KWFURE
KWFUSY		W	Same as wfusys in mks units	/PRINTW/(*	)	PRINTW	W	KWFUSY
KWFUTK		W	Same as wfutk in mks units	/PRINTW/(*	)	PRINTW	W	KWFUTK
KWFUTR		W	Same as wfutrp in mks units	/PRINTW/(*	)	PRINTW	W	KWFUTR
KWFU2		W	Same as wfu2(1) in mks units	/PRINTW/(*	)	PRINTW	W	KWFU2
KWGASP		W	Same as wgaspr in mks units	/PRINTW/(*	)	PRINTW	W	KWGASP
KWGNAV		W	Same as wgnav in mks units	/PRINTW/(*	)	PRINTW	W	KWGNAV
KWGROS		W	Same as wgross in mks units	/PRINTW/(*	)	PRINTW	W	KWGROS
KWHORZ		W	Same as whorz in mks units	/PRINTW/(*	)	PRINTW	W	KWHORZ
KWHYCA		W	Same as whycad in mks units	/PRINTW/(*	)	PRINTW	W	KWHYCA
KWINFU		W	Same as winfut in mks units	/PRINTW/(*	)	PRINTW	W	KWINFU
KWINOX		W	Same as winoxt in mks units	/PRINTW/(*	)	PRINTW	W	KWINOX
KWINST		W	Same as winstk in mks units	/PRINTW/(*	)	PRINTW	W	KWINST

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
KWINSU		W	Same as winsul in mks units	/PRINTW/(*	)	PRINTW	W	KWINSU
KWINSI		W	Same as winst in mks units	/PRINTW/(*	)	PRINTW	W	KWINSI
KWLANC		W	Same as wlanch in mks units	/PRINTW/(*	)	PRINTW	W	KWLANC
KWLG		W	Same as wlg in mks units	/PRINTW/(*	)	PRINTW	W	KWLG
KWLOSS		W	Same as wloss in mks units	/PRINTW/(*	)	PRINTW	W	KWLOSS
KWLRD		W	Same as wlrd in mks units	/PRINTW/(*	)	PRINTW	W	KWLRD
KWNACE		W	Same as wnacel in mks units	/PRINTW/(*	)	PRINTW	W	KWNACE
KWOIL		W	Same as woil in mks units	/PRINTW/(*	)	PRINTW	W	KWOIL
KWOILR		W	Same as woilrs in mks units	/PRINTW/(*	)	PRINTW	W	KWOILR
KWORSU		W	Same as worsul in mks units	/PRINTW/(*	)	PRINTW	W	KWORSU
KWOXID		W	Same as woxid in mks units	/PRINTW/(*	)	PRINTW	W	KWOXID
KWOXLO		W	Same as woxlos in mks units	/PRINTW/(*	)	PRINTW	W	KWOXLO
KWOXRE		W	Same as woxres in mks units	/PRINTW/(*	)	PRINTW	W	KWOXRE
KWOXSY		W	Same as woxsys in mks units	/PRINTW/(*	)	PRINTW	W	KWOXSY
KWOXTK		W	Same as woxtk in mks units	/PRINTW/(*	)	PRINTW	W	KWOXTK
KWOXTR		W	Same as woxtrp in mks units	/PRINTW/(*	)	PRINTW	W	KWOXTR
KWOX2		W	Same as wox2(1) in mks units	/PRINTW/(*	)	PRINTW	W	KWOX2
KWPASS		W	Same as wpass in mks units	/PRINTW/(*	)	PRINTW	W	KWPASS
KWPAYL		W	Same as wpayl in mks units	/PRINTW/(*	)	PRINTW	W	KWPAYL
KWPERS		W	Same as wpers in mks units	/PRINTW/(*	)	PRINTW	W	KWPERS
KWPOWC		W	Same as wpowcd in mks units	/PRINTW/(*	)	PRINTW	W	KWPOWC
KWPOWE		W	Same as wpower in mks units	/PRINTW/(*	)	PRINTW	W	KWPOWE
KWPOWF		W	Same as wpowfo in mks units	/PRINTW/(*	)	PRINTW	W	KWPOWF
KWPOWR		W	Same as wpowrs in mks units	/PRINTW/(*	)	PRINTW	W	KWPOWR
KWPOWT		W	Same as wpowtk in mks units	/PRINTW/(*	)	PRINTW	W	KWPOWT
KWPPRO		W	Same as wpprov in mks units	/PRINTW/(*	)	PRINTW	W	KWPPRO
KWPREI		W	Same as wpreig in mks units	/PRINTW/(*	)	PRINTW	W	KWPREI
KWPROP		W	Same as wprop in mks units	/PRINTW/(*	)	PRINTW	W	KWPROP
KWPRSY		W	Same as wprsys in mks units	/PRINTW/(*	)	PRINTW	W	KWPRSY
KWRESI		W	Same as wresid in mks units	/PRINTW/(*	)	PRINTW	W	KWRESI
KWRESR		W	Same as wresrv in mks units	/PRINTW/(*	)	PRINTW	W	KWRESR
KWSORC		W	Same as wsorce in mks units	/PRINTW/(*	)	PRINTW	W	KWSORC
KWSRTR		W	Same as wsrtrp in mks units	/PRINTW/(*	)	PRINTW	W	KWSRTR
KWSTAB		W	Same as wstab in mks units	/PRINTW/(*	)	PRINTW	W	KWSTAB
KWSURF		W	Same as wsurf in mks units	/PRINTW/(*	)	PRINTW	W	KWSURF
KWTHRS		W	Same as wthrst in mks units	/PRINTW/(*	)	PRINTW	W	KWTHRS
KWTO		W	Same as wto in mks units	/PRINTW/(*	)	PRINTW	W	KWTO
KWTPS		W	Same as wtps in mks units	/PRINTW/(*	)	PRINTW	W	KWTPS
KWVERT		W	Same as wvert in mks units	/PRINTW/(*	)	PRINTW	W	KWVERT
KWWET		W	Same as wwet in mks units	/PRINTW/(*	)	PRINTW	W	KWWET
KWWING		W	Same as wwing in mks units	/PRINTW/(*	)	PRINTW	W	KWWING
PLDLOS		W	Payload loss (lb) input as c(296)	/PRINTW/(*	)	PRINTW	W	PLDLOS
PLLOS		W	Same as pldlos in mks units	/PRINTW/(*	)	PRINTW	W	PLLOS
PRINTW		E	Subroutine to print vehicle weight data	/PRINTW/($	)	PRINTW WTVOL	E S	PRINTW PRINTW
WABFTK		I	Weight of air breathing propulsion system tanks	/WTCALC/(	2)	PRINTW WTSCH	I M	WABFTK WABFTK
WABFU		I	Weight of jp fuel	/WTCALC/(	3)	PRINTW PRWTSM TAMPER WTSCH	I M I M	WABFU WABFU WABFU WABFU

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WABPR		I	Weight of air breathing engines	/WTCALC/(	4)	PRINTW WTSCH	I M	WABPR WABPR
WACRES		I	Weight of attitude control fuel reserve	/WTCALC/(	5)	PRINTW STORE WTSCH	I M M	WACRES WACRES WACRES
WACS		I	Weight of attitude control system	/WTCALC/(	6)	PRINTW WTSCH	I M	WACS WACS
WACSFO		I	Weight of attitude control fuel plus oxidizer	/WTCALC/(	7)	PRINTW STORE WTSCH	I M M	WACSFO WACSFO WACSFO
WACSTK		I	Weight of attitude control tankage	/WTCALC/(	8)	PRINTW WTSCH	I M	WACSTK WACSTK
WAERO		I	Weight of aerodynamic controls	/WTCALC/(	9)	PRINTW WTSCH	I M	WAERO WAERO
WAUXT		I	Weight of separation system	/WTCALC/(	10)	PRINTW STORE WTSCH	I I M	WAUXT WAUXT WAUXT
WBASIC		I	Total weight of basic body	/WTCALC/(	11)	PRINTW PROTHR WTSCH	I I M	WBASIC WBASIC WBASIC
WBODY		I	Total weight of body group	/WTCALC/(	12)	PRINTW WTSCH	I M	WBODY WBODY
WBOO		I	Booster gross weight	/JUMPY /(	3)	PRINTW TAMPER WTVOL	I I O	WBOO WBOO WBOO
WCARGO		I	Payload weight or cargo	/WTCALC/(	14)	PRINTW WTSCH	I M	WCARGO WCARGO
WCOMM		I	Communication system weight	/WTCALC/(	15)	PRINTW WTSCH	I M	WCOMM WCOMM
WCONT		I	Contingency and growth weight	/WTCALC/(	16)	PRINTW TAMPER WTSCH	I I M	WCONT WCONT WCONT
WCOVER		I	Total weight of thermal protection system cover panels	/WTCALC/(	17)	PRINTW WTSCH	I M	WCOVER WCOVER
WDECAY		I	Thrust decay propellant weight	/WTCALC/(	18)	PRINTW STORE WTSCH	I M M	WDECAY WDECAY WDECAY
WDOCK		I	Docking structure weight	/WTCALC/(	21)	PRINTW WTSCH	I M	WDOCK WDOCK
WDPLOY		I	Deployable aerodynamic device weight	/WTCALC/(	22)	PRINTW WTSCH	I M	WDPLOY WDPLOY
WDRY		I	Stage dry weight	/WTCALC/(	24)	PRINTW TAMPER WTSCH	I I M	WDRY WDRY WDRY
WENGS		I	Weight of rocket engines installed	/WTCALC/(	28)	PRINTW WTSCH	I M	WENGS WENGS
WENGS2		I	Weight of secondary engines	/WTCALC/(	29)	PRINTW WTSCH	I M	WENGS2 WENGS2
WFAIR		I	Weight of fairings and shrouds	/WTCALC/(	30)	PRINTW WTSCH	I M	WFAIR WFAIR
WFROST		I	Frost and ice weight	/WTCALC/(	33)	PRINTW STORE WTSCH	I M M	WFROST WFROST WFROST
WFUL		I	Fuel weight	/WTCALC/(	43)	PRINTW WTSCH	I M	WFUL WFUL
WFULOS		I	Vented fuel	/WTCALC/(	44)	PRINTW STORE WTSCH	I M M	WFULOS WFULOS WFULOS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WFUOX		I	Weight of main and secondary propellant	/WTCALC/(	46)	PRINTW	I	WFUOX
						TAMPER	I	WFUOX
						WTSCH	M	WFUOX
						WTVOL	I	WFUOX
WFURES		I	Fuel reserve	/WTCALC/(	47)	PRINTW	I	WFURES
						STORE	M	WFURES
						TAMPER	I	WFURES
						WTSCH	M	WFURES
WFUSYS		I	Total fuel system weight	/WTCALC/(	48)	PRINTW	I	WFUSYS
						WTSCH	M	WFUSYS
WFUTK		I	Wt of non-structural fuel tankage	/WTCALC/(	49)	PRINTW	I	WFUTK
						WTSCH	M	WFUTK
WFUTK2		I	Wt of secondary fuel tank and system	/WTCALC/(	50)	PRINTW	I	WFUTK2
						WTSCH	M	WFUTK2
WFUTRP		I	Trapped fuel weight	/WTCALC/(	52)	PRINTW	I	WFUTRP
						STORE	M	WFUTRP
						WTSCH	M	WFUTRP
WFU2		I	Weight of secondary fuel	/WTCALC/(	34)	PRINTW	I	WFU2
						TAMPER	I	WFU2
						WTSCH	M	WFU2
WGASPR		I	Weight of gas and pressurant	/WTCALC/(	53)	PRINTW	I	WGASPR
						STORE	M	WGASPR
						WTSCH	M	WGASPR
WGNAV		I	Guidance and navigation system wt	/WTCALC/(	54)	PRINTW	I	WGNAV
						WTSCH	M	WGNAV
WGROSS		I	Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
						PRWTSM	I	WGROSS
						SOLVE	M	WGROSS
						STORE	M	WGROSS
						TAMPER	I	WGROSS
						WTSCH	M	WGROSS
						WTVOL	I	WGROSS
WHORZ		I	Horizontal stabilizer wt.	/WTCALC/(	55)	PRINTW	I	WHORZ
						PROTHR	I	WHORZ
						WTSCH	M	WHORZ
WHYCAD		I	Hydraulic / pneumatic system wt	/WTCALC/(	56)	PRINTW	I	WHYCAD
						WTSCH	M	WHYCAD
WINFUT		I	Weight of intergral fuel tank	/WTCALC/(	57)	PRINTW	I	WINFUT
						PROTHR	I	WINFUT
						WTSCH	M	WINFUT
WINOXT		I	Weight of integral oxidizer tank	/WTCALC/(	58)	PRINTW	I	WINOXT
						PROTHR	I	WINOXT
						WTSCH	M	WINOXT
WINST		I	Weight of instrument system	/WTCALC/(	60)	PRINTW	I	WINST
						WTSCH	M	WINST
WINSTK		I	Total weight of tank insulation	/WTCALC/(	59)	PRINTW	I	WINSTK
						WTSCH	M	WINSTK
WLANCH		I	Launch gear weight	/WTCALC/(	68)	PRINTW	I	WLANCH
						WTSCH	M	WLANCH
WLG		I	Landing gear and controls weight	/WTCALC/(	69)	PRINTW	I	WLG
						WTSCH	M	WLG
WLOSS		I	In-flight weight loss	/WTCALC/(	70)	PRINTW	I	WLOSS
						PRWTSM	I	WLOSS
						WTSCH	O	WLOSS
WLRD		I	Launch and recovery system weight	/WTCALC/(	71)	PRINTW	I	WLRD
						WTSCH	M	WLRD
WNACEL		I	Pylons, nacel,and pod weights	/WTCALC/(	72)	PRINTW	I	WNACEL
						WTSCH	M	WNACEL
WOIL		I	Service item losses	/WTCALC/(	74)	PRINTW	I	WOIL
						WTSCH	M	WOIL
WOILRS		I	Service item reserves	/WTCALC/(	75)	PRINTW	I	WOILRS
						WTSCH	M	WOILRS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WORSUL		I	Orientation,control, and separation system weight	/WTCALC/(	76)	PRINTW WTSCH	I M	WORSUL WORSUL
WOXID		I	Main impulse oxidizer weighter	/WTCALC/(	87)	PRINTW WTSCH	I M	WOXID WOXID
WOXLOS		I	Vented oxidizer	/WTCALC/(	88)	PRINTW STORE WTSCH	I M M	WOXLOS WOXLOS WOXLOS
WOXRES		I	Oxidizer reserve	/WTCALC/(	89)	PRINTW STORE TAMPER WTSCH	I M I M	WOXRES WOXRES WOXRES WOXRES
WOXSYS		I	Oxidizer system weight	/WTCALC/(	90)	PRINTW WTSCH	I M	WOXSYS WOXSYS
WOXTK		I	Non-sturctural tank wt.- oxidizer	/WTCALC/(	91)	PRINTW WTSCH	I M	WOXTK WOXTK
WOXTK2		I	Secondary system oxidizer tank wt	/WTCALC/(	92)	PRINTW WTSCH	I M	WOXTK2 WOXTK2
WOXTRP		I	Trapped oxidizer weight	/WTCALC/(	94)	PRINTW STORE WTSCH	I M M	WOXTRP WOXTRP WOXTRP
WOX2		I	Secondary oxidizer weight	/WTCALC/(	84)	PRINTW TAMPER WTSCH	I I M	WOX2 WOX2 WOX2
WPASS		I	Weight of passengers	/WTCALC/(	96)	PRINTW WTSCH	I M	WPASS WPASS
WPAYL		I	Payload weight	/WTCALC/(	97)	PRINTW TAMPER WTSCH	I I M	WPAYL WPAYL WPAYL
WPERS		I	Crew gear and life support weight	/WTCALC/(	98)	PRINTW WTSCH	I M	WPERS WPERS
WPOWCD		I	Power conditioning equipment wt.	/WTCALC/(	99)	PRINTW WTSCH	I M	WPOWCD WPOWCD
WPOWFO		I	Power system propellant wt.	/WTCALC/(	101)	PRINTW STORE WTSCH	I M M	WPOWFO WPOWFO WPOWFO
WPOWRS		I	Power system propellant reserve	/WTCALC/(	102)	PRINTW STORE WTSCH	I M M	WPOWRS WPOWRS WPOWRS
WPOWTK		I	Prime power system tank weight	/WTCALC/(	103)	PRINTW WTSCH	I M	WPOWTK WPOWTK
WPPROV		I	Personnel provisions	/WTCALC/(	104)	PRINTW WTSCH	I M	WPPROV WPPROV
WPREIG		I	Pre-ignition losses	/WTCALC/(	105)	PRINTW PRWTSM WTSCH	I M M	WPREIG WPREIG WPREIG
WPROP		I	Total weight- propulsion group	/WTCALC/(	106)	PRINTW WTSCH	I M	WPROP WPROP
WPRSYS		I	Pressurization system weight	/WTCALC/(	107)	PRINTW WTSCH	I M	WPRSYS WPRSYS
WRESID		I	Weight of residuals	/WTCALC/(	109)	PRINTW WTSCH	I M	WRESID WRESID
WRESRV		I	Propellant reservs	/WTCALC/(	110)	PRINTW WTSCH	I M	WRESRV WRESRV
WSECST		I	Secondary body structure wt	/WTCALC/(	112)	PRINTW PROTHR WTSCH	I I M	WSECST WSECST WSECST
WSORCE		I	Prime power system weight	/WTCALC/(	113)	PRINTW WTSCH	I M	WSORCE WSORCE
WSRTRP		I	Trapped oxidizer weight	/WTCALC/(	114)	PRINTW STORE WTSCH	I M M	WSRTRP WSRTRP WSRTRP

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
WSTAB		I	Engine gimbal system wt	/WTCALC/(	115)	PRINTW	I	WSTAB
						WTSCH	M	WSTAB
WSURF		I	Aero surface wt	/WTCALC/(	116)	PRINTW	I	WSURF
						WTSCH	M	WSURF
WTHRST		I	Thrust structure wt	/WTCALC/(	118)	PRINTW	I	WTHRST
						WTSCH	M	WTHRST
WTO		I	Take- off weight	/WTCALC/(	119)	PRINTW	I	WTO
						WTSCH	M	WTO
WTPS		I	Induced environmental protection wt	/WTCALC/(	120)	PRINTW	I	WTPS
						WTSCH	M	WTPS
WVERT		I	Vertical fin weight	/WTCALC/(	121)	PRINTW	I	WVERT
						PROTHR	I	WVERT
						WTSCH	M	WVERT
WWET		I	Operating weight-empty	/WTCALC/(	132)	PRINTW	I	WWET
						WTSCH	M	WWET
WWING		I	Total structural wt. Of wing	/WTCALC/(	133)	PRINTW	I	WWING
						PROTHR	I	WWING
						WTSCH	M	WWING
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE PRINTW                                                 PRINTW
C                                                                       PRINTW
C   PRINTW - PRINTS WEIGHTS                                             PRINTW
C                                                                       PRINTW
      REAL KWSTAB                                                       PRINTW
      REAL KWABPR,KWABFT,KWABFU                                         PRINTW
     *,KILO ,KWSURF,KWWING,KWVERT,KWHORZ,KWFAIR,KWBODY,KWINFU,KWBASI,   PRINTW
     1KWSECS,KWTPS ,KWCOVE,KWINSU,KWLG  ,KWPROP,KWENGS,KWNACE,KWFUTK,   PRINTW
     2KWOXTK,KWINST,KWFUSY,KWOXSY,KWPRSY,KWAERO,KWPOWE,KWAPU ,KWPOWT,   PRINTW
     3KWELCA,KWHYCA,KWGNAV,KWPPRO,KWDRY ,KWCONT,KWPERS,KWPAYL,KWCARG,   PRINTW
     4KWPASS,KWRESI,KWGASP,KWFUTR,KWOXTR,KWSRTR,KWRESR,KWFURE,          PRINTW
     *KWOXRE,KWPOWR,KWOILR,KWACRE,       KWWET ,KWLOSS,KWFULO,KWOXLO,   PRINTW
     *KWPOWF,KWOIL ,KWDECA,KWFUOX,KWFUL ,KWPREI,KWGROS                  PRINTW
     *,KFUTK2,KOXTK2                                                    PRINTW
      REAL  KWLRD,KWLANC,KWDPLO,KWDOCK,KWORSU,KWAUXT,KWACS              PRINTW
     *,KWACSF,KWSORC,KWINS1,KWFU2,KWOX2                                 PRINTW
     1,KWINOX,KWTHRS,KWENG2,KWPOWC,KWTO,KWOXID,KWACST,KWBOO             PRINTW
      COMMON / JUMPY / JUMP, WBIG, WBOO                                 PRINTW
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS   ,ANTANK   ,ASRATO   ,ASWEEP   ,C(300)   ,CBBODY   ,CFUEL(6),   CINPUT
     2CHBODY   ,CLBODY   ,CSBODY   ,CSFAIR   ,CSFUTK   ,CSHORZ   ,CSOXTK   ,   CINPUT
     3CSPLAN   ,CSVERT   ,CSWING   ,CTHRST   ,CTHST2   ,DEF(5)   ,FXWOVS   ,   CINPUT
     4ISP(6)   ,ITPS     ,K(30)    ,KIN      ,LF       ,MR(6)    ,NCREW    ,   CINPUT
     5NENGS    ,NLISTO   ,NPASS    ,NWL      ,PCHAM    ,Q        ,RHOFU    ,   CINPUT
     6RHOFU2   ,RHOX     ,RHOX2    ,SBODY    ,TOL      ,TOVERC   ,TPRATO   ,   CINPUT
     7TYTAIL   ,VBODY    ,WGROSS                                        CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN    ,CTIP     ,GAL      ,GSPAN    ,   VOLCAL
     2HBODY    ,LBODY    ,RTOD     ,SFAIR    ,SFUTK    ,SHORZ    ,SOXTK    ,   VOLCAL
     3SPLAN    ,STPS(1)  ,SVERT    ,SWING    ,SXPOS    ,TDEL     ,TROOT    ,   VOLCAL
     4TTOT     ,TTOT2    ,TTOTAL   ,VBODYA   ,VBODY1   ,VBODY2   ,VCARGO   ,   VOLCAL
     5VCREW    ,VFUTK    ,VFUTK2   ,VINSTK   ,VLGBAY   ,VOTHER   ,VOXTK    ,   VOLCAL
     6VOXTK2   ,VPROP    ,VSTRUC                                        VOLCAL
      COMMON/WTCALC/     ABFSYS   ,WABFTK   ,WABFU    ,WABPR    ,WACRES   ,   WTCALC
     1WACS     ,WACSFO   ,WACSTK   ,WAERO    ,WAUXT    ,WBASIC   ,WBODY    ,   WTCALC
     2WBPUMP   ,WCARGO   ,WCOMM    ,WCONT    ,WCOVER   ,WDECAY   ,WDIST1   ,   WTCALC
     3WDIST2   ,WDOCK    ,WDPLOY   ,WORANS   ,WDRY     ,WELCAD   ,WEMPTY   ,   WTCALC
     4WENGMT   ,WENGS    ,WENGS2   ,WFAIR    ,WFCONT   ,WFDCAY   ,WFROST   ,   WTCALC
     5WFU2(3)  ,WFUEL(6),WFUL     ,WFULOS   ,WFUNCT   ,WFUOX    ,WFURES   ,   WTCALC
     6WFUSYS   ,WFUTK    ,WFUTK2   ,WFUTOT   ,WFUTRP   ,WGASPR   ,WGNAV    ,   WTCALC
     7WHORZ    ,WHYCAD   ,WINFUT   ,WINOXT   ,WINSTK   ,WINST    ,WINSUL   ,   WTCALC
     8WJET(6)  ,WLANCH   ,WLG      ,WLOSS    ,WLRD     ,WNACEL   ,WODCAY   ,   WTCALC
     9WOIL     ,WOILRS   ,WORSUL   ,WOVERS   ,WOX(6)   ,WOX2(3)  ,WOXID    ,   WTCALC
     1WOXLOS   ,WOXRES   ,WOXSYS   ,WOXTK    ,WOXTK2   ,WOXTOT   ,WOXTRP   ,   WTCALC
     2WP       ,WPASS    ,WPAYL    ,WPERS    ,WPOWCO   ,WPOWER   ,WPOWFO   ,   WTCALC
     3WPOWRS   ,WPOWTK   ,WPPROV   ,WPREIG   ,WPROP    ,WPRSYS   ,WREFUL   ,   WTCALC
     4WRESIO   ,WRESRV   ,WSEAL    ,WSECST   ,WSORCE   ,WSRTRP   ,WSTAB    ,   WTCALC
     5WSURF    ,WTABC    ,WTHRST   ,WTO      ,WTPS     ,WVERT    ,WWAIT(10),   WTCALC
     6WWET     ,WWING    ,WZROFU   ,WABTRP   ,WABRES   ,WMNOTP   ,WMNFTP   ,   CKOUT
     7WMNORS   ,WMNFRS   ,WACOTP   ,WACFTP   ,WPWOTP   ,WPWFTP   ,WGAS     ,   CKOUT
     8WABFUC   ,WACORS   ,WACFRS   ,WPWORS   ,WPWFRS                      CKOUT
      KILO = 0.4535924                                                  PRINTW
      KWABPR = WABPR * KILO                                             PRINTW
      KWABFT = WABFTK * KILO                                            PRINTW
      KWABFU = WABFU * KILO                                             PRINTW
      KWSTAB=KILO*WSTAB                                                 PRINTW
      KWSURF = KILO*WSURF                                               PRINTW
      KWWING = KILO*WWING                                               PRINTW
      KWVERT = KILO*WVERT                                               PRINTW
      KWHORZ = KILO*WHORZ                                               PRINTW
      KWFAIR = KILO*WFAIR                                               PRINTW
      KWACSF = KILO*WACSFO                                              PRINTW
      KWBODY = KILO*WBODY                                               PRINTW
      KWBASI = KILO*WBASIC                                              PRINTW
      KWSECS = KILO*WSECST                                              PRINTW
      KWTPS  = KILO*WTPS                                                PRINTW
      KWCOVE = KILO*WCOVER                                              PRINTW
      KWDPLO=KILO*WDPLOY                                                PRINTW
      KWINSU = KILO*WINSUL                                              PRINTW
      KWDOCK = KILO * WDOCK                                             PRINTW
      KWLG   = KILO*WLG                                                 PRINTW
      KWPROP = KILO*WPROP                                               PRINTW
      KWGROS = KILO*WGROSS                                              PRINTW
```

```
      KWTO   = KILO* WTO                                                 PRINTW
      KWENGS = KILO*WENGS                                                PRINTW
      KWNACE = KILO*WNACEL                                               PRINTW
      KWFUTK = KILO*WFUTK                                                PRINTW
      KWOXTK = KILO*WOXTK                                                PRINTW
      KWINST = KILO*WINSTK                                               PRINTW
      KWFUSY = KILO*WFUSYS                                               PRINTW
      KWOXSY = KILO*WOXSYS                                               PRINTW
      KWACS = KILO*WACS                                                  PRINTW
      KWAUXT = KILO* WAUXT                                               PRINTW
      KWPRSY = KILO*WPRSYS                                               PRINTW
      KWSORC = KILO*WSORCE                                               PRINTW
      KWAERO = KILO*WAERO                                                PRINTW
      KWPOWE = KILO*WPOWER                                               PRINTW
      KWPOWT = KILO*WPOWTK                                               PRINTW
      KWELCA = KILO*WELCAD                                               PRINTW
      KWHYCA = KILO*WHYCAD                                               PRINTW
      KWGNAV = KILO*WGNAV                                                PRINTW
      KWINS1 = KILO*WINST                                                PRINTW
      KWCOMM = KILO*WCOMM                                                PRINTW
      KWPPRO = KILO*WPPROV                                               PRINTW
      KWLRD = KILO*WLRD                                                  PRINTW
      KWLANC = KILO*WLANCH                                               PRINTW
      KWDRY  = KILO*WDRY                                                 PRINTW
      KWCONT = KILO*WCONT                                                PRINTW
      KWPERS = KILO* WPERS                                               PRINTW
      KWPAYL = KILO*WPAYL                                                PRINTW
      KWCARG = KILO*WCARGO                                               PRINTW
      KWPASS = KILO*WPASS                                                PRINTW
      KWRESI = KILO*WRESID                                               PRINTW
      KWGASP = KILO*WGASPR                                               PRINTW
      KWFUTR = KILO*WFUTRP                                               PRINTW
      KWOXTR = KILO*WOXTRP                                               PRINTW
      KWSRTR = KILO*WSRTRP                                               PRINTW
      KWRESR = KILO*WRESRV                                               PRINTW
      KWFURE = KILO*WFURES                                               PRINTW
      KWOXRE = KILO*WOXRES                                               PRINTW
      KWPOWR = KILO*WPOWRS                                               PRINTW
      KWOILR = KILO*WOILRS                                               PRINTW
      KWACRE = KILO*WACRES                                               PRINTW
      KWWET  = KILO*WWET                                                 PRINTW
      KWLOSS = KILO*WLOSS                                                PRINTW
      FROST=KILO*WFROST                                                  PRINTW
      KWFULO = KILO*WFULOS                                               PRINTW
      KWOXLO = KILO*WOXLOS                                               PRINTW
      KFUTK2=KILO*WFUTK2                                                 PRINTW
      KOXTK2=KILO*WOXTK2                                                 PRINTW
      KWPOWF = KILO*WPOWFO                                               PRINTW
      KWOIL  = KILO*WOIL                                                 PRINTW
      KWDECA = KILO*WDECAY                                               PRINTW
      KWFUOX = KILO*WFUOX                                                PRINTW
      KWFUL  = KILO*WFUL                                                 PRINTW
      KWOXID = KILO*WOXID                                                PRINTW
      KWFU2 = KILO * WFU2(1)                                             PRINTW
      KWOX2 = KILO * WOX2(1)                                             PRINTW
      KWORSU = KILO*WORSUL                                               PRINTW
      KWACST = KILO*WACSTK                                               PRINTW
      KWPREI = KILO*WPREIG                                               PRINTW
      KWINFU = KILO * WINFUT                                             PRINTW
      KWINOX =KILO*WINOXT                                                PRINTW
      KWTHRS =KILO*WTHRST                                                PRINTW
      KWENG2 =KILO*WENGS2                                                PRINTW
      KWPOWC =KILO*WPOWCD                                                PRINTW
      KWACRE =KILO*WACRES                                                PRINTW
      PLDLOS=C(296)                                                      PRINTW
      PLLOS=KILO*PLDLOS                                                  PRINTW
      WRITE(6,66)                                                        PRINTW
   66 FORMAT (1H1)                                                       PRINTW
   67 CONTINUE                                                           PRINTW
      IF(JUMP.EQ.1) GO TO 45                                             PRINTW  95
      WRITE(6,25)                                                        PRINTW
   25 FORMAT (1H0,32HWEIGHT BREAKDOWN - ORBITAL STAGE,33X6HPOUNDS,20X    PRINTW
     *   9HKILOGRAMS)                                                    PRINTW
      GO TO 50                                                           PRINTW  50
   45 WRITE (6,26)                                                       PRINTW
```

1489

```
      26 FORMAT (1HO,31HWEIGHT BREAKDOWN - BOOST STAGE ,33X6HPOUNDS,20X       PRINTW
     *   9HKILOGRAMS)                                                         PRINTW
      50 WRITE(6,100) WSURF,KWSURF                                            PRINTW
     150 WRITE(6,200) WWING,KWWING                                            PRINTW
 250    WRITE (6,13500) WHORZ,KWHORZ                                          PRINTW
     350 WRITE(6,400)WVERT,KWVERT                                             PRINTW
     650 WRITE(6,700)WFAIR,KWFAIR                                             PRINTW
     750 WRITE(6,800)WBODY,KWBODY                                             PRINTW
         WRITE(6,900) WINFUT,KWINFU                                           PRINTW
         WRITE(6,1000) WINOXT,KWINOX                                          PRINTW
 1050    WRITE(6,1100)WBASIC,KWBASI                                           PRINTW
         WRITE(6,1300) WSECST,KWSECS                                          PRINTW
 1350    WRITE (6,1400) WTHRST,KWTHRS                                         PRINTW
    1550 WRITE(6,1600) WTPS,KWTPS                                             PRINTW
    1750 WRITE(6,1800) WCOVER,KWCOVE                                          PRINTW
    1850 WRITE(6,1900) WINSUL,KWINSU                                          PRINTW
    2150 WRITE(6,2200)WLRD,KWLRD                                              PRINTW
    2250 WRITE(6,2300)WLANCH  ,KWLANC                                         PRINTW
    2350 WRITE(6,2400)    WDPLOY,KWDPLO                                       PRINTW
    2450 WRITE(6,2500)    WLG,KWLG                                            PRINTW
    2550 WRITE(6,2600) WDOCK,KWDOCK                                           PRINTW
    2750 WRITE(6,2800) WPROP,KWPROP                                           PRINTW
    2850 WRITE(6,2900) WENGS,KWENGS                                           PRINTW
         WRITE(6,2870) WENGS2,KWENG2                                          PRINTW
         WRITE(6,3300) WFUTK,KWFUTK                                           PRINTW
         WRITE(6,3400) WOXTK,KWOXTK                                           PRINTW
         WRITE(6,3410) WFUTK2,KFUTK2                                          PRINTW
         WRITE(6,3420) WOXTK2,KOXTK2                                          PRINTW
         WRITE(6,3500) WINSTK,KWINST                                          PRINTW
    3550 WRITE(6,3600) WFUSYS,KWFUSY                                          PRINTW
         WRITE(6,3700) WOXSYS,KWOXSY                                          PRINTW
         WRITE(6,3800) WPRSYS,KWPRSY                                          PRINTW
         WRITE(6,4110) WABPR,KWABPR                                           PRINTW
         WRITE(6,3100) WNACEL,KWNACE                                          PRINTW
         WRITE(6,4115) WABFTK,KWABFT                                          PRINTW
    4250 WRITE(6,4300)WORSUL,KWORSU                                           PRINTW
    4350 WRITE(6,4400)  WAUXT,KWAUXT                                          PRINTW
         WRITE(6,4470) WSTAB,KWSTAB                                           PRINTW
 4450    WRITE(6,4500)  WAERO,KWAERO                                          PRINTW
    4550 WRITE(6,4600) WACS,KWACS                                             PRINTW
         WRITE(6,4800) WACSTK,KWACST                                          PRINTW
    4850 WRITE(6,4900)WPOWER,KWPOWE                                           PRINTW
 4950    WRITE(6,5000) WSORCE,KWSORC                                          PRINTW
         WRITE(6,5300) WPOWTK,KWPOWT                                          PRINTW
 5350    WRITE (6,5400)WPOWCD,KWPOWC                                          PRINTW
    5550 WRITE(6,5600)WHYCAD,KWHYCA                                           PRINTW
    5650 WRITE(6,5700) WGNAV,KWGNAV                                           PRINTW
    5750 WRITE(6,5800) WINST,KWINS1                                           PRINTW
    5850 WRITE(6,5900) WCOMM,KWCOMM                                           PRINTW
    6550 WRITE(6,6600)WPPROV,KWPPRO                                           PRINTW
    7250 WRITE(6,7300) WDRY,KWDRY                                             PRINTW
         WRITE(6,7400) WCONT,KWCONT                                           PRINTW
 7450    WRITE(6,7500) WPERS,KWPERS                                           PRINTW
    7750 WRITE(6,7800) WPAYL  ,KWPAYL                                         PRINTW
    7851 WRITE (6,7900) WPASS, KWPASS                                         PRINTW
         WRITE (6,7860) WCARGO, KWCARG                                        PRINTW
    8350 WRITE(6,8400)WRESID,KWRESI                                           PRINTW
         WRITE(6,8500) WGASPR,KWGASP                                          PRINTW
    8550 WRITE(6,8600)WFUTRP,KWFUTR                                           PRINTW
         WRITE(6,8700) WOXTRP,KWOXTR                                          PRINTW
    8850 WRITE(6,8900)WSRTRP,KWSRTR                                           PRINTW
    8950 WRITE(6,9000)WRESRV,KWRESR                                           PRINTW
    9050 WRITE(6,9100)WFURES,KWFURE                                           PRINTW
         WRITE(6,9200) WOXRES,KWOXRE                                          PRINTW
         WRITE(6,9300) WPOWRS,KWPOWR                                          PRINTW
         WRITE(6,9400) WOILRS,KWOILR                                          PRINTW
         WRITE(6,9500) WACRES,KWACRE                                          PRINTW
    9650 WRITE(6,9700)WWET,KWWET                                              PRINTW
    9750 WRITE(6,9800)WLOSS,KWLOSS                                            PRINTW
         WRITE(6,9801) PLDLOS,PLLOS                                           PRINTW
         WRITE(6,9898) WFROST,FROST                                           PRINTW
    9850 WRITE(6,9900)WFULOS,KWFULO                                           PRINTW
    9950 WRITE(6,10000)WOXLOS,KWOXLO                                          PRINTW
   10050 WRITE(6,10100)WPOWFO,KWPOWF                                          PRINTW
   10150 WRITE(6,10200)  WACSFO,KWACSF                                        PRINTW
```

```
      WRITE(6,10400) WOIL,KWOIL                                         PRINTW
      WRITE(6,10410) WABFU,KWABFU                                       PRINTW
      WRITE(6,10600)WDECAY,KWDECA                                       PRINTW
10650 WRITE(6,10700)WFUOX,KWFUOX                                        PRINTW
10750 WRITE(6,10800)WFUL,KWFUL                                          PRINTW
      WRITE(6,11400) WOXID,KWOXID                                       PRINTW
      WRITE(6,11500)WFU2(1),KWFU2                                       PRINTW
      WRITE(6,11600)WOX2(1),KWOX2                                       PRINTW
      WRITE(6,11900)WTO,KWTO                                            PRINTW
      WRITE(6,12300) WPREIG,KWPREI                                      PRINTW
      WRITE(6,12500) WGROSS,KWGROS                                      PRINTW
      IF(JUMP.EQ.0) GO TO 9999                                          PRINTW  9999
      KWBOO = WBOO * KILO                                               PRINTW
      WRITE(6,13010) WBOO , KWBOO                                       PRINTW
 9999 CONTINUE                                                          PRINTW
      RETURN                                                            PRINTW
100   FORMAT(1H0,52HAERODYNAMIC SURFACES                                PRINTW
     *F18.0,F27.0    )                                                  PRINTW
  200 FORMAT(1H ,52H    WING + WING MOUNTED CONTROL SURFACES            PRINTW
     *F9.0,F27.0)                                                       PRINTW
  400 FORMAT(1H ,52H    VERTICAL SURFACES                               PRINTW
     *F9.0,F27.0)                                                       PRINTW
  700 FORMAT(1H ,52H    FAIRINGS,SHROUDS AND ASSOCIATED STRUCTURE       PRINTW
     *F9.0,F27.0)                                                       PRINTW
  800 FORMAT(1H0,52HBODY STRUCTURE                                      PRINTW
     *F18.0,F27.0    )                                                  PRINTW
  900 FORMAT (1H ,50H    STRUCTURAL FUEL CONTAINERS                     PRINTW
     *F11.0,F27.0)                                                      PRINTW
 1000 FORMAT (1H ,50H    STRUCTURAL OXIDIZER CONTAINERS                 PRINTW
     *F11.0,F27.0)                                                      PRINTW
1100  FORMAT(1H ,52H    BASIC BODY STRUCTURE                            PRINTW
     *F9.0,F27.0)                                                       PRINTW
 1300 FORMAT(1H ,52H    SECONDARY STRUCTURE                             PRINTW
     *F9.0,F27.0)                                                       PRINTW
 1400 FORMAT(1H ,52H    THRUST STRUCTURE                                PRINTW
     *F9.0,F27.0)                                                       PRINTW
 1600 FORMAT(1H0,52HINDUCED ENVIRONMENTAL PROTECTION                    PRINTW
     *F18.0,F27.0    )                                                  PRINTW
 1800 FORMAT(1H ,52H    COVER PANELS,NON-STRUCTURAL                     PRINTW
     *F9.0,F27.0)                                                       PRINTW
1900  FORMAT(1H ,52H    VEHICLE INSULATION                              PRINTW
     *F9.0,F27.0)                                                       PRINTW
 2200 FORMAT(1H0,52HLAUNCH,RECOVERY AND DOCKING                         PRINTW
     *F18.0,F27.0    )                                                  PRINTW
 2300 FORMAT(1H ,52H    LAUNCH GEAR                                     PRINTW
     *F9.0,F27.0)                                                       PRINTW
 2400 FORMAT(1H ,52H    DEPLOYABLE AERODYNAMIC DEVICES                  PRINTW
     *F9.0,F27.0)                                                       PRINTW
2500  FORMAT(1H ,52H    ALIGHTING GEAR                                  PRINTW
     *F9 0,F27.0)                                                       PRINTW
 2600 FORMAT(1H ,52H    DOCKING STRUCTURE                               PRINTW
     *F9.0,F27.0)                                                       PRINTW
 2800 FORMAT(1H0,52HPROPULSION                                          PRINTW
     *F18.0,F27.0    )                                                  PRINTW
2870  FORMAT(1H ,52H    SECONDARY ENGINES AND ACCESSORIES               PRINTW
     *F9.0,F27.0)                                                       PRINTW
 2900 FORMAT(1H ,52H    ENGINES AND ACCESSORIES                         PRINTW
     *F9.0,F27.0)                                                       PRINTW
 3100 FORMAT(1H ,52H    NACELLES,PODS,PYLONS,SUPPORTS                   PRINTW
     *F9.0,F27.0)                                                       PRINTW
 3300 FORMAT(1H ,52H    FUEL CONTAINERS AND SUPPORTS (NON-STRUCTURAL)   PRINTW
     *F9.0,F27.0)                                                       PRINTW
 3400 FORMAT(1H ,52H    OXIDIZER CONTAINERS AND SUPPORTS(NON-STRUCTURAL) PRINTW
     *F9.0,F27.0)                                                       PRINTW
3410  FORMAT(1H ,52H    SECONDARY FUEL TANKAGE AND SYSTEMS              PRINTW
     *  F9.0,F27.0)                                                     PRINTW
3420  FORMAT(1H ,52H    SECONDARY OXIDIZER TANKAGE AND SYSTEMS          PRINTW
     *    F9.0,F27.0)                                                   PRINTW
 3500 FORMAT(1H ,52H    PROPELLANT INSULATION                           PRINTW
     *F9.0,F27.0)                                                       PRINTW
 3600 FORMAT(1H ,52H    FUEL SYSTEM - MAIN                              PRINTW
     *F9.0,F27.0)                                                       PRINTW
 3700 FORMAT(1H ,52H    OXIDIZER SYSTEM - MAIN                          PRINTW
     *F9.0,F27.0)                                                       PRINTW
3800  FORMAT(1H ,52H    PURGE SYSTEMS                                   PRINTW
```

```
       *F9.0,F27.0)                                                    PRINTW
   4110 FORMAT(1H ,52H    AIRBREATHING ENGINES AND INSTALLATION        PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4115 FORMAT(1H ,52H    AIRBREATHING PROPULSION TANKAGE AND SYSTEMS  PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4300 FORMAT(1H0,52HORIENTATION,SEPARATION AND ULLAGE CONTROL        PRINTW
       *F18.0,F27.0    )                                               PRINTW
   4400 FORMAT(1H ,52H    SEPARATION SYSTEMS                           PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4470 FORMAT(1H ,52H    STABILITY AND CONTROL, ENGINE GIMBAL         PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4500 FORMAT(1H ,52H    AERODYNAMIC CONTROLS                         PRINTW
       *F9.0,F27.0)                                                    PRINTW
  4600  FORMAT(1H ,52H    SPATIAL ATTITUDE CONTROL SYSTEM              PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4800 FORMAT(1H ,52H    CONTROL PROPELLANT TANKAGE AND SYSTEMS       PRINTW
       *F9.0,F27.0)                                                    PRINTW
   4900 FORMAT(1H0,52HPRIME POWER SOURCE AND DISTRIBUTION              PRINTW
       *F18.0,F27.0    )                                               PRINTW
  5000  FORMAT(1H ,52H    POWER SOURCE UNITS                           PRINTW
       *F9.0,F27.0)                                                    PRINTW
   5300 FORMAT(1H ,52H    POWER SOURCE TANKAGE AND SYSTEMS             PRINTW
       *F9.0,F27.0)                                                    PRINTW
   5400 FORMAT(1H0,52HPOWER CONVERSION AND DISTRIBUTION                PRINTW
       *F18.0,F27.0    )                                               PRINTW
   5600 FORMAT(1H ,52H    HYDRAULIC/PNEUMATIC                          PRINTW
       *F9.0,F27.0)                                                    PRINTW
   5700 FORMAT(1H0,52HGUIDANCE AND NAVIGATION                          PRINTW
       *F18.0,F27.0    )                                               PRINTW
   5800 FORMAT(1H0,52HINSTRUMENTATION                                  PRINTW
       *F18.0,F27.0    )                                               PRINTW
   5900 FORMAT(1H0,52HCOMMUNICATION                                    PRINTW
       *F18.0,F27.0    )                                               PRINTW
   6600 FORMAT(1H0,52HPERSONNEL PROVISIONS                             PRINTW
       *F18.0,F27.0    )                                               PRINTW
   7300 FORMAT(1H0,52HDRY STRUCTURE                                    PRINTW
       *, 7X,1H(,F10.0,1H),17X,1H(,F10.0,1H)        )                  PRINTW
  7400  FORMAT(1H0,52HDESIGN RESERVE (CONTINGENCY)                     PRINTW
       *F18.0,F27.0    )                                               PRINTW
  7500  FORMAT(1H0,52HPERSONNEL                                        PRINTW
       *  F18.0,F27.0)                                                 PRINTW
   7800 FORMAT(1H0,52HPAYLOAD                                          PRINTW
       *F18.0,F27.0    )                                               PRINTW
   7860 FORMAT(1H ,52H    MISSION EQUIPMENT/PAYLOAD                    PRINTW
       *F9.0,F27.0)                                                    PRINTW
   7900 FORMAT(1H ,52H    CARGO OR UPPER STAGE                         PRINTW
       *F9.0,F27.0)                                                    PRINTW
   8400 FORMAT(1H0,52HRESIDUAL PROPELLANT AND SERVICE ITEMS            PRINTW
       *F18.0,F27.0    )                                               PRINTW
  8500  FORMAT(1H ,52H    TANK PRESSURIZATION AND PURGE GASES          PRINTW
       *F9.0,F27.0)                                                    PRINTW
  8600  FORMAT(1H ,52H    TRAPPED FUEL                                 PRINTW
       *F9.0,F27.0)                                                    PRINTW
  8700  FORMAT(1H ,52H    TRAPPED OXIDIZER                             PRINTW
       *F9.0,F27.0)                                                    PRINTW
   8900 FORMAT(1H ,52H    SERVICE ITEMS RESIDUALS                      PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9000 FORMAT(1H0,52HRESERVE PROPELLANT AND SERVICE ITEMS             PRINTW
       *F18.0,F27.0    )                                               PRINTW
   9100 FORMAT(1H ,52H    FUEL-MAIN PROPULSION                         PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9200 FORMAT(1H ,52H    OXIDIZER-MAIN PROPULSION                     PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9300 FORMAT(1H ,52H    POWER SOURCE PROPELLANTS                     PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9400 FORMAT(1H ,52H    LUBRICANTS                                   PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9500 FORMAT(1H ,52H    ATTITUDE CONTROL PROPELLANTS                 PRINTW
       *F9.0,F27.0)                                                    PRINTW
   9700 FORMAT(1H0,52HOPERATING WEIGHT EMPTY                           PRINTW
       *, 7X,1H(,F10.0,1H),17X,1H(,F10.0,1H)        )                  PRINTW
   9800 FORMAT(1H0,52HIN-FLIGHT LOSSES                                 PRINTW
       *F18.0,F27.0    )                                               PRINTW
   9801 FORMAT(1H ,52H    HUCKEY  PUCK                                 PRINTW
       *F9.0,F27.0  )                                                  PRINTW
```

1492

```
  9898 FORMAT(1H ,52H    ICE AND FROST                                   PRINTW
     *F9.0,F27.0)                                                        PRINTW
 9900  FORMAT(1H ,52H    FUEL VENTED                                     PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10000 FORMAT(1H ,52H    OXIDIZER VENTED                                 PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10100 FORMAT(1H ,52H    POWER SOURCE PROPELLANTS                        PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10200 FORMAT(1H ,52H    ATTITUDE CONTROL PROPELLANTS                    PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10400 FORMAT(1H ,52H    LUBRICANTS AND OTHER SERVICE ITEMS              PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10410 FORMAT(1H ,52H    FLYBACK FUEL                                    PRINTW
     *F9.0,F27.0)                                                        PRINTW
 10600 FORMAT(1H0,52HTHRUST DECAY PROPELLANTS                            PRINTW
     *F18.0,F27.0    )                                                   PRINTW
 10700 FORMAT(1H0,52HPROPELLANTS                                         PRINTW
     *F18.0,F27.0    )                                                   PRINTW
 10800 FORMAT(1H ,52H    FUEL - MAIN                                     PRINTW
     *F9.0,F27.0)                                                        PRINTW
 11400 FORMAT(1H ,52H    OXIDIZER - MAIN                                 PRINTW
     *F9.0,F27.0)                                                        PRINTW
 11500 FORMAT(1H ,52H    FUEL - SECONDARY                                PRINTW
     *F9.0,F27.0)                                                        PRINTW
 11600 FORMAT(1H ,52H    OXIDIZER - SECONDARY                            PRINTW
     *F9.0,F27.0)                                                        PRINTW
 11900 FORMAT(1H0,52HMASS AT FULL THRUST                                 PRINTW
     *F18.0,F27.0    )                                                   PRINTW
 12300 FORMAT(1H0,52HPRE-IGNITION LOSSES                                 PRINTW
     *F18.0,F27.0    )                                                   PRINTW
 12500 FORMAT(1H0,52HMAXIMUM GROSS WEIGHT                                PRINTW
     *, 7X,1H(,F10.0,1H),15X,1H(,F10.0,1H)          )                    PRINTW
 13010 FORMAT(1H0,50HGROSS WEIGHT OF ONE BOOSTER                         PRINTW
     *  2XF18.0,F27.0)                                                   PRINTW
 13500 FORMAT(1H ,52H    HORIZONTAL SURFACES                             PRINTW
     * F9.0,F27.0)                                                       PRINTW
       END                                                               PRINTW
```

SUBRØUTINE PRITEQ

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
C		I	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
						PRITEQ	I	C
						PRITVA	I	C
						STORE	M	C
						WTSCH	I	C
						WTVOL	O	C
I		W	Do loop counter	/PRITEQ/(*	)	PRITEQ	W	I
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHOF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

PRITEQ

```
      SUBROUTINE PRITEQ                                                 PRITEQ
C                                                                       PRITEQ
C         SUBROUTINE TO PRINT WEIGHT AND VOLUME COEFFICIENTS            PRITEQ
C                                                                       PRITEQ
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,       VV(3),     QP(14),    EROR,      PZ(5),     VQ,       SW(20), SIZING
     *SV(28),   SQ(37,5),  SE(11),    TLAT,      TLNG,                  SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,      WLOO,      DWEB,      DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,   NNB,         WEO,       WEB,       WO,       WLO,    SIZING
     *DVO,       DVB,       MUB,       MUO,       VSTG,      WPO            SIZING
     *,JTYP,     BECO      BSTG       ORBI       ITNBW    ITNOW   ,         SIZING
     * SVOPSQ   , SVOCON   ,IHUNT     ,IOPSTG   ,ISZO(19)                   UN
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS   ,ANTANK   ,ASRATO   ,ASWEEP   ,C(300)   ,CBBODY   ,CFUEL(6),  CINPUT
     2CHBODY   ,CLBODY   ,CSBODY   ,CSFAIR   ,CSFUTK   ,CSHORZ   ,CSOXTK   ,  CINPUT
     3CSPLAN   ,CSVERT   ,CSWING   ,CTHRST   ,CTHST2   ,DEF(5)   ,FXWOVS   ,  CINPUT
     4ISP(6)   ,ITPS     ,K(30)    ,KIN      ,LF       ,MR(6)    ,NCREW    ,  CINPUT
     5NENGS    ,NLISTO   ,NPASS    ,NWL      ,PCHAM    ,Q        ,RHOFU    ,  CINPUT
     6RHOFU2   ,RHOX     ,RHOX2    ,SBODY    ,TOL      ,TOVERC   ,TPRATO   ,  CINPUT
     7TYTAIL   ,VBODY    ,WGROSS                                            CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP     ,GAL      ,GSPAN    ,  VOLCAL
     2HBODY    ,LBODY    ,RTOD     ,SFAIR    ,SFUTK    ,SHORZ    ,SOXTK    ,  VOLCAL
     3SPLAN    ,STPS(1)  ,SVERT    ,SWING    ,SXPOS    ,TDEL     ,TROOT    ,  VOLCAL
     4TTOT     ,TTOT2    ,TTOTAL   ,VBODYA   ,VBODY1   ,VBODY2   ,VCARGO   ,  VOLCAL
     5VCREW    ,VFUTK    ,VFUTK2   ,VINSTK   ,VLGBAY   ,VOTHER   ,VOXTK    ,  VOLCAL
     6VOXTK2   ,VPROP    ,VSTRUC                                            VOLCAL
      COMMON/WTCALC/      ABFSYS   ,WABFTK   ,WABFU    ,WABPR    ,WACRES   ,  WTCALC
     1WACS     ,WACSFO   ,WACSTK   ,WAERO    ,WAUXT    ,WBASIC   ,WBODY    ,  WTCALC
     2WBPUMP   ,WCARGO   ,WCOMM    ,WCONT    ,WCOVER   ,WDECAY   ,WDIST1   ,  WTCALC
     3WDIST2   ,WDOCK    ,WDPLOY   ,WDRANS   ,WDRY     ,WELCAD   ,WEMPTY   ,  WTCALC
     4WENGMT   ,WENGS    ,WENGS2   ,WFAIR    ,WFCONT   ,WFDCAY   ,WFROST   ,  WTCALC
     5WFU2(3)  ,WFUEL(6),WFUL      ,WFULOS   ,WFUNCT   ,WFUOX    ,WFURES   ,  WTCALC
     6WFUSYS   ,WFUTK    ,WFUTK2   ,WFUTOT   ,WFUTRP   ,WGASPR   ,WGNAV    ,  WTCALC
     7WHORZ    ,WHYCAD   ,WINFUT   ,WINOXT   ,WINSTK   ,WINST    ,WINSUL   ,  WTCALC
     8WJET(6)  ,WLANCH   ,WLG      ,WLOSS    ,WLRD     ,WNACEL   ,WODCAY   ,  WTCALC
     9WOIL     ,WOILRS   ,WORSUL   ,WOVERS   ,WOX(6)   ,WOX2(3)  ,WOXID    ,  WTCALC
     1WOXLOS   ,WOXRES   ,WOXSYS   ,WOXTK    ,WOXTK2   ,WOXTOT   ,WOXTRP   ,  WTCALC
     2WP       ,WPASS    ,WPAYL    ,WPERS    ,WPOWCO   ,WPOWER   ,WPOWFO   ,  WTCALC
     3WPOWRS   ,WPOWTK   ,WPPROV   ,WPREIG   ,WPROP    ,WPRSYS   ,WREFUL   ,  WTCALC
     4WRESID   ,WRESRV   ,WSEAL    ,WSECST   ,WSORCE   ,WSRTRP   ,WSTAB    ,  WTCALC
     5WSURF    ,WTABC    ,WTHRST   ,WTO      ,WTPS     ,WVERT    ,WWAIT(10), WTCALC
     6WWET     ,WWING    ,WZROFU   ,WABTRP   ,WABRES   ,WMNOTP   ,WMNFTP   ,  CKOUT
     7WMNORS   ,WMNFRS   ,WACOTP   ,WACFTP   ,WPWOTP   ,WPWFTP   ,WGAS     ,  CKOUT
     8WABFUC   ,WACORS   ,WACFRS   ,WPWORS   ,WPWFRS                          CKOUT
  900 WRITE(6,1150)                                                     PRITEQ
 1150 FORMAT(1H1,28HWEIGHT COEFFICIENTS = C(I), /)                      PRITEQ
      DO 300 I = 1,300                                                  PRITEQ
      IF(C(I).NE.0.) WRITE (6,1200) I,C(I)                              PRITEQ
  300 CONTINUE                                                          PRITEQ
 1200 FORMAT(I5,F13.5)                                                  PRITEQ
      WRITE(6,1250)                                                     PRITEQ
 1250 FORMAT(1H0,38HVOLUME COEFFICIENTS = K(I), I = 1,30 =/)            PRITEQ
      WRITE(6,1300)(I,K(I),I=1,30)                                      PRITEQ
 1300 FORMAT( 5(I5,F13.5))                                              PRITEQ
      CALL PRITVA                                                       PRITEQ
      RETURN                                                            PRITEQ
       END                                                              PRITEQ
```

SUBRØUTINE
PRITVA

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
C		I	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
						PRITEQ	I	C
						PRITVA	I	C
						STORE	M	C
						WTSCH	I	C
						WTVOL	O	C
CBBODY		I	Body width coeff.	/CINPUT/(	305)	PRITVA	I	CBBODY
						STORE	M	CBBODY
						WTSCH	I	CBBODY
CHBODY		I	Body height or coeff	/CINPUT/(	312)	PRITVA	I	CHBODY
						STORE	M	CHBODY
						WTSCH	I	CHBODY
CLBODY		I	Body length or coeff	/CINPUT/(	313)	PRITVA	I	CLBODY
						STORE	M	CLBODY
						WTSCH	I	CLBODY
CSBODY		I	Total body wetted area or coeff	/CINPUT/(	314)	PRITVA	I	CSBODY
						STORE	M	CSBODY
						WTSCH	I	CSBODY
CSFAIR		I	Fairing planform area or coeff	/CINPUT/(	315)	PRITVA	I	CSFAIR
						STORE	M	CSFAIR
						WTSCH	I	CSFAIR
CSFUTK		I	Fuel tank surface area or coeff	/CINPUT/(	316)	PRITVA	I	CSFUTK
						STORE	M	CSFUTK
						WTSCH	I	CSFUTK
CSHORZ		I	Horizontal stabalizer planform area	/CINPUT/(	317)	PRITVA	I	CSHORZ
						STORE	M	CSHORZ
						WTSCH	I	CSHORZ
CSOXTK		I	Oxidizer tank surface area coeff	/CINPUT/(	318)	PRITVA	I	CSOXTK
						STORE	M	CSOXTK
						WTSCH	I	CSOXTK
CSPLAN		I	Body planform area or coeff	/CINPUT/(	319)	PRITVA	I	CSPLAN
						STORE	M	CSPLAN
						WTSCH	I	CSPLAN
CSVERT		I	Vertical fin planform area or coeff	/CINPUT/(	320)	PRITVA	I	CSVERT
						STORE	M	CSVERT
						WTSCH	I	CSVERT
CTHRST		I	Vac. Thrust-to-weight ratio	/CINPUT/(	322)	PRITVA	I	CTHRST
						STORE	M	CTHRST
						WTSCH	I	CTHRST
						WTVOL	M	CTHRST
CTHST2		I	Secondary propulsion t/w	/CINPUT/(	323)	PRITVA	I	CTHST2
						STORE	M	CTHST2
						WTSCH	I	CTHST2
I		W	Do loop counter	/PRITVA/(*	)	PRITVA	W	I
JUMP		I	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH	O	JUMP
						PRINTW	I	JUMP
						PRITVA	I	JUMP
						PRWTSM	M	JUMP
						WTSCH	I	JUMP
						WTVOL	M	JUMP
NCREW		I	Number of crew members	/CINPUT/(	375)	PRITVA	I	NCREW
						STORE	M	NCREW
						WTSCH	I	NCREW
NENGS		I	Total number engines per stage	/CINPUT/(	376)	PRITVA	I	NENGS
						STORE	M	NENGS
						WTSCH	I	NENGS
RHOFU		I	Fuel density	/CINPUT/(	382)	PRITVA	I	RHOFU
						STORE	M	RHOFU
						WTSCH	I	RHOFU
RHOFU2		I	Secondary fuel desnity	/CINPUT/(	383)	PRITVA	I	RHOFU2
						STORE	M	RHOFU2
						WTSCH	I	RHOFU2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
RHOX		I	Oxidizer density	/CINPUT/(	384)	PRITVA STORE WTSCH	I M I	RHOX RHOX RHOX
RHOX2		I	Secondary oxidizer density	/CINPUT/(	385)	PRITVA STORE WTSCH	I M I	RHOX2 RHOX2 RHOX2
SE		I	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP PRITVA SIZEMR SUMOUT TAMPER THRUST VEHDF WTSCH WTVOL	O I I I M I M I M	SE SE SE SE SE SE SE SE SE
SQ		I	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM FLYBKP ISPRAT PDBC PRITVA RANGE REU3 SIZE SIZEMR SIZIN STAU SUMOUT TAMPAR TAMPER THRUST TRTOSZ VEHDF WTVOL	M M I I I M O O M M I M O M M M M M	SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ
TTOT		I	Total stage vac. Thrust	/VOLCAL/(	21)	PRITVA STORE WTSCH WTVOL	I I M I	TTOT TTOT TTOT TTOT
TTOT2		I	Total stage vac. Secondary thrust	/VOLCAL/(	22)	PRITVA WTSCH	I M	TTOT2 TTOT2
WJET		I	Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA PRWTSM STORE TAMPER WTSCH	I M O I M	WJET WJET WJET WJET WJET
WWAIT		I	Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA PRWTSM STORE TAMPER WTSCH	I I M I M	WWAIT WWAIT WWAIT WWAIT WWAIT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
.UN06.		0	File of all output data	/.UN06./(6	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAY02	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      SUBROUTINE PRITVA                                                  PRITVA
C                                                                        PRITVA
C  PRITVA - PRINTS VARIABLES                                             PRITVA
C                                                                        PRITVA
      COMMON / JUMPY / JUMP, WBIG, WBOO                                  PRITVA
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                            SIZING
      COMMON /SIZING/                                                    SIZING
C          PHASE II SIZING PARAMERERS                                    SIZING
     *TZ,     VV(3),    QP(14),   EROR,     PZ(5),     VQ,      SW(20),  SIZING
     *SV(28), SQ(37,5), SE(11),   TLAT,     TLNG,                        SIZING
C          PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,    WLOO,     DWEB,     DWEO,     TOLWT,    WPB,      TWRAT2,  SIZING
     *BK1,    BK2,      BK3,      BK4,      ISIZE,    TRAFLG,   TWRATO,  SIZING
     *OK1,    OK2,      OK3,      OK4,      PRFLG,    IPASS,    IPSMAX,  SIZING
     *AEXIT,  TVACO,    NO,       WFO,      IOVEL,    ISPO,     ISPB,    SIZING
     *XPL,    TVACB,    NNB,      WEO,      WEB,      WO,       WLO,     SIZING
     *DVO,     DVB,      MUB,      MUO,      VSTG,     WPO               SIZING
     *,JTYP,   BECO ,    BSTG ,    ORBI ,   ITNBW ,  ITNOW  ,            SIZING
     * SVDPSQ  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZO(19)                     UH
      REAL KIN                                                           CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                 CINPUT
      REAL NENGS                                                         CINPUT
      COMMON/CINPUT/                                                     CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(360)  ,CBBODY  ,CFUEL(6),    CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,    CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,    CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,    CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,    CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,    CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                           CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,    VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,    VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TOEL    ,TROOT   ,    VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,    VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,    VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                           VOLCAL
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,    WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,    WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,    WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,    WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,    WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUDX   ,WFURES  ,    WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,    WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,    WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,    WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,    WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,    WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFO  ,    WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,    WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,    WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),   WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,    CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,    CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                         CKOUT
      WRITE (6,26001)                                                    PRITVA
      WRITE (6,1000) CBBODY                                              PRITVA
      WRITE (6,2000) CHBODY                                              PRITVA
      WRITE (6,3000) CLBODY                                              PRITVA
      WRITE (6,4000) CSBODY                                              PRITVA
      WRITE (6,6000) CSFAIR                                              PRITVA
      WRITE( 6, 7000) CSFUTK                                             PRITVA
      WRITE( 6, 8000) CSHORZ                                             PRITVA
      WRITE( 6, 9000) CSOXTK                                             PRITVA
      WRITE( 6,10000) CSPLAN                                             PRITVA
      WRITE( 6,11000) CSVERT                                             PRITVA
      WRITE (6,13000) CTHRST                                             PRITVA
      WRITE(6,13002) CTHST2                                              PRITVA
      WRITE(6,16000) NCREW                                               PRITVA
      WRITE(6,17000) NENGS                                               PRITVA
      WRITE(6,20000) RHOFU                                               PRITVA
      WRITE(6,21000) RHOFU2                                              PRITVA
      WRITE(6,22000) RHOX                                                PRITVA
      WRITE(6,22010) RHOX2                                               PRITVA
      WRITE(6,24000) CTHRST,WWAIT(3),C(129),NENGS,TTOT                   PRITVA
      WRITE(6,25000) CTHST2,WWAIT(6),C(158),TTOT2                        PRITVA
```

```
  100 WRITE (6,26001)                                                   PRITVA
      WRITE (6,26000)(I,WJET(I),I=1,6)                                  PRITVA
      IF(JUMP.EQ.0) GO TO 200                                           PRITVA  200
      WRITE (6,1500) SW(12)                                             PRITVA
      WRITE (6,1505) SW(14)                                             PRITVA
      WRITE (6,1510) SW(6)                                              PRITVA
      WRITE (6,1515) SW(11)                                             PRITVA
      WRITE (6,1520) SW(15)                                             PRITVA
      WRITE(6,1525) SQ(10,2)                                            PRITVA
      WRITE(6,30000) SE(6)                                              PRITVA
       WRITE(6,30001) SQ(5,1)                                           PRITVA
      WRITE(6,30002) SQ(6,1)                                            PRITVA
      WRITE(6,30003)  SQ(9,1)                                           PRITVA
      WRITE(6,30004) SQ(7,1)                                            PRITVA
      WRITE(6,30005) SQ(8,1)                                            PRITVA
      IF(JUMP.EQ.1) WRITE (6,999)                                       PRITVA
  200 RETURN                                                            PRITVA
  999 FORMAT (1H1)                                                      PRITVA
 1000 FORMAT (1H0,41HCBBODY = WIDTH OF BODY / VBODY ** .3333 =,F9.4)    PRITVA
 1500 FORMAT (28H0SPECIFIC FUEL CONSUMPTION = F7.5)                     PRITVA
 1505 FORMAT (28H CRUISE BACK VELOCITY      = F9.2,8H FT/SEC )          PRITVA
 1510 FORMAT (28H THRUST RATIO              = F8.5)                     PRITVA
 1515 FORMAT (28H L/D                       = F8.5)                     PRITVA
 1520 FORMAT (28H FLYBACK RANGE             = F9.2,7H N. MI.)           PRITVA
 1525 FORMAT(29H BANK ANGLE DURING REENTRY = F6.2,4H DEG)               PRITVA
 2000 FORMAT (1H ,41HCHBODY = HEIGHT OF BODY / VBODY** .3333 =,F9.4)    PRITVA
 3000 FORMAT (1H ,41HCLBODY = LENGTH OF BODY / VBODY** .3333 =,F9.4)    PRITVA
 4000 FORMAT (1H ,47HCSBODY = WETTED AREA OF BODY / VBODY ** .6667 =,   PRITVA
     * F9.4)                                                            PRITVA
 6000 FORMAT (1H ,42HCSFAIR = FAIRING AREA / BODY WETTED AREA =,F9.4)   PRITVA
 7000 FORMAT (1H ,58HCSFUTK = WETTED AREA OF FUEL TANKS / TANK VOLUME ** PRITVA
     *.6667 =,F9.4)                                                     PRITVA
 8000 FORMAT(1H ,43HCSHORZ = STABILIZER AREA/THEO WING AREA=      ,F9.4) PRITVA
 9000 FORMAT (1H ,63HCSOXTK = WETTED AREA OF OXIDIZER TANKS / TANK VOLUM PRITVA
     *E ** .6667 =,F9.4)                                                PRITVA
10000 FORMAT (1H ,54HCSPLAN = VEHICLE PLANFORM AREA / BODY VOLUME **.666 PRITVA
     *7 =,F9.4)                                                         PRITVA
11000 FORMAT (1H ,57HCSVERT = AREA OF EACH VERTICAL SURFACE / VBODY ** . PRITVA
     *6667 =,F9.4)                                                      PRITVA
13000 FORMAT (1H ,30HCTHRST = ENGINE THRUST / WTO =,F9.4)               PRITVA
13002 FORMAT(47H1CTHST2 = AIRBREATHING ENGINE THRUST / WENTRY =,F9.4)   PRITVA
16000 FORMAT (1H0,24HNCREW = NUMBER OF CREW =,F5.0)                     PRITVA
17000 FORMAT (1H ,27HNENGS = NUMBER OF ENGINES =,F4.0)                  PRITVA
20000 FORMAT (1H ,22HRHOFU = FUEL DENSITY =,F6.2)                       PRITVA
21000 FORMAT (1H ,43HRHOFU2= SECONDARY PROPULSION FUEL DENSITY =,F6.2)  PRITVA
22000 FORMAT (1H ,25HRHOX = OXIDIZER DENSITY =,F6.2)                    PRITVA
22010 FORMAT (1H ,45HRHOX2 = SECONDARY PROPULSION OXIDIZER DENSITY,F6.2) PRITVA
24000 FORMAT (1H ,66HTTOT = MAIN PROPULSION THRUST = CTHRST * WORBIT + C PRITVA
     *(129) * NENGS =  / 10X                                            PRITVA
     *                    F6.3,3H * ,F7.0, 3H + ,F8.0,3H * ,F2.0,3H = ,F7.0) PRITVA
25000 FORMAT (1H0,64HTTOT2 = AIRBREATHING ENGINE THRUST = CTHST2 X WETUR PRITVA
     *N  C(158) = *  / 10X F6.3,3H * ,F7.0,3H + ,F7.0,3H = ,F7.0)       PRITVA
26000 FORMAT (1H ,5HWJET(,I1,3H) =,F10.1)                               PRITVA
26001 FORMAT(1H0)                                                       PRITVA
30000 FORMAT(1H0,42HMAXIMUM ASCENT DYNAMIC PRESSURE          = F8.2,    PRITVA
     14H PSF)                                                           PRITVA
30001 FORMAT(1H ,42HTIME AT MAX Q                            = F8.3,    PRITVA
     14H SEC)                                                           PRITVA
30002 FORMAT(1H ,42HALTITUDE AT MAX Q                        = F8.0,    PRITVA
     13H FT)                                                            PRITVA
30003 FORMAT(1H ,42HMACH NUMBER AT MAX Q                     = F7.3)    PRITVA
30004 FORMAT(1H ,42HRELATIVE VELOCITY AT MAX Q               = F9.2,    PRITVA
     14H FPS)                                                           PRITVA
30005 FORMAT(1H ,42HRELATIVE FLIGHT PATH ANGLE AT MAX Q      = F6.2,    PRITVA
     14H DEG)                                                           PRITVA
       END                                                              PRITVA
```

20 OCT 72 6.01-46

SUBRØUTINE
PRØTHR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
BBODY		I	Body width used by set0 to set common to zero	/VOLCAL/(	1)	PROTHR SETO WTSCH	I O M	BBODY O BBODY
CROOT		I	Ing root chord	/VOLCAL/(	2)	PROTHR WTSCH	I M	CROOT CROOT
CSPAN		I	Structural span along 0.5 chord	/VOLCAL/(	3)	PROTHR WTSCH	I M	CSPAN CSPAN
HBODY		I	Body height	/VOLCAL/(	7)	PROTHR WTSCH	I O	HBODY HBODY
KBBODY		W	Same as bbody in mks units	/PROTHR/(*	)	PROTHR	W	KBBODY
KCROOT		W	Same as croot in mks units	/PROTHR/(*	)	PROTHR	W	KCROOT
KCSPAN		W	Same as cspan in mks units	/PROTHR/(*	)	PROTHR	W	KCSPAN
KHBODY		W	Same as hbody in mks units	/PROTHR/(*	)	PROTHR	W	KHBODY
KILO		W	Pounds to kilogram conversion	/PROTHR/(*	)	PROTHR	W	KILO
KLBODY		W	Same as lbody in mks units	/PROTHR/(*	)	PROTHR	W	KLBODY
KLROOT		O	Same as croot in mks units	/PROTHR/(*	)	PROTHR	O	KLROOT
KSBODY		W	Same as sbody in mks units	/PROTHR/(*	)	PROTHR	W	KSBODY
KSHORZ		W	Same as shorz in mks units	/PROTHR/(*	)	PROTHR	W	KSHORZ
KSPLAN		W	Same as splan in mks units	/PROTHR/(*	)	PROTHR	W	KSPLAN
KSVERT		W	Same as svert in mks units	/PROTHR/(*	)	PROTHR	W	KSVERT
KSWING		W	Same as swing in mks units	/PROTHR/(*	)	PROTHR	W	KSWING
KSXPOS		W	Same as sxpos in mks units	/PROTHR/(*	)	PROTHR	W	KSXPOS
KUWBOD		W	Same as uwbody in mks units	/PROTHR/(*	)	PROTHR	W	KUWBOD
KUWHOR		W	Same as uwhorz in mks units	/PROTHR/(*	)	PROTHR	W	KUWHOR
KUWVER		W	Same as uwvert in mks units	/PROTHR/(*	)	PROTHR	W	KUWVER
KUWWIN		W	Same as uwwing in mks units	/PROTHR/(*	)	PROTHR	W	KUWWIN
KWOVER		W	Same as wovers in mks units	/PROTHR/(*	)	PROTHR	W	KWOVER
LBODY		I	Body length	/VOLCAL/(	8)	PROTHR TAMPER WTSCH	I I M	LBODY LBODY LBODY
PROTHR		E	Subroutine to print misc data	/PROTHR/($	)	PRINTV PROTHR	S E	PROTHR PROTHR
SBODY		I	Total body wetted area	/CINPUT/(	386)	PROTHR TAMPER WTSCH	I I M	SBODY SBODY SBODY
SFUTK		I	Total fuel tank wetted area	/VOLCAL/(	11)	PROTHR WTSCH	I M	SFUTK SFUTK
SFUTKK		W	Same as sfutk in mks units	/PROTHR/(*	)	PROTHR	W	SFUTKK
SHORZ		I	Horizontal stabilizer planform area	/VOLCAL/(	12)	PROTHR WTSCH	I M	SHORZ SHORZ
SOXTK		I	Total oxidizer tank wetted area	/VOLCAL/(	13)	PROTHR WTSCH	I M	SOXTK SOXTK
SOXTKK		W	Same as soxtk in mks units	/PROTHR/(*	)	PROTHR	W	SOXTKK
SPLAN		I	Body planform area	/VOLCAL/(	14)	PROTHR TAMPER WTSCH WTVOL	I I O I	SPLAN SPLAN SPLAN SPLAN
SVERT		I	Vertical fin planform area	/VOLCAL/(	16)	PROTHR WTSCH	I M	SVERT SVERT
SWING		I	Gross wing area	/VOLCAL/(	17)	PROTHR STORE WTSCH	I M M	SWING SWING SWING
SXPOS		I	Exposed wing area	/VOLCAL/(	18)	PROTHR WTSCH	I O	SXPOS SXPOS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
TOVERC		I	Wing thickness over choord ratio	/CINPUT/(	388)	PROTHR	I	TOVERC
						STORE	M	TOVERC
						WTSCH	I	TOVERC
UWBODY		W	Body unit weight	/PROTHR/(*	)	PROTHR	W	UWBODY
UWHORZ		W	Horizontal tail unit weight	/PROTHR/(*	)	PROTHR	W	UWHORZ
UWVERT		W	Vertical tail unit weight	/PROTHR/(*	)	PROTHR	W	UWVERT
UWWING		W	Wing unit weight	/PROTHR/(*	)	PROTHR	W	UWWING
WBASIC		I	Total weight of basic body	/WTCALC/(	11)	PRINTW	I	WBASIC
						PROTHR	I	WBASIC
						WTSCH	M	WBASIC
WHORZ		I	Horizontal stabilizer wt.	/WTCALC/(	55)	PRINTW	I	WHORZ
						PROTHR	I	WHORZ
						WTSCH	M	WHORZ
WINFUT		I	Weight of intergral fuel tank	/WTCALC/(	57)	PRINTW	I	WINFUT
						PROTHR	I	WINFUT
						WTSCH	M	WINFUT
WINOXT		I	Weight of integral oxidizer tank	/WTCALC/(	58)	PRINTW	I	WINOXT
						PROTHR	I	WINOXT
						WTSCH	M	WINOXT
WOVERS		I	Wing loading	/WTCALC/(	77)	PROTHR	I	WOVERS
						TAMPER	I	WOVERS
						WTSCH	M	WOVERS
WSECST		I	Secondary body structure wt	/WTCALC/(	112)	PRINTW	I	WSECST
						PROTHR	I	WSECST
						WTSCH	M	WSECST
WVERT		I	Vertical fin weight	/WTCALC/(	121)	PRINTW	I	WVERT
						PROTHR	I	WVERT
						WTSCH	M	WVERT
WWING		I	Total structural wt. Of wing	/WTCALC/(	133)	PRINTW	I	WWING
						PROTHR	I	WWING
						WTSCH	M	WWING

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
.UN06.		0	File of all output data	/.UN06./(	)	BLICO	0	.UN06
						BNDRYC	0	.UN06
						CRASH	0	.UN06
						FRENCH	0	.UN06
						FXDAT	0	.UN06
						GEINP	0	.UN06
						HUNT	0	.UN06
						INEDIT	0	.UN06
						ITER8	0	.UN06
						MODELA	0	.UN06
						MOMJ	0	.UN06
						MPSI	0	.UN06
						OUT	0	.UN06
						PAYO2	0	.UN06
						PRINT	0	.UN06
						PRINTV	0	.UN06
						PRINTW	0	.UN06
						PRITEQ	0	.UN06
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      SUBROUTINE PROTHR                                                 PROTHR
C                                                                       PROTHR
C   PROTHR - PRINT  ALL OTHER QUANTITIES                                PROTHR
C                                                                       PROTHR
      REAL KILO, KSBODY, KSWING, KWOVER, KSVERT, KSHORZ, KUWWIN,        PROTHR
     1 KUWVER, KUWHOR, KUWBOD, KLROOT, KLBODY, KBBODY, KHBODY,          PROTHR
     2 KSXPOS, KSPLAN                                                   PROTHR
      REAL KCSPAN,KCROOT                                                PROTHR
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS   ,ANTANK   ,ASRATO   ,ASWEEP   ,C(300)   ,CBBODY   ,CFUEL(6),  CINPUT
     2CHBODY   ,CLBODY   ,CSBODY   ,CSFAIR   ,CSFUTK   ,CSHORZ   ,CSOXTK   ,  CINPUT
     3CSPLAN   ,CSVERT   ,CSWING   ,CTHRST   ,CTHST2   ,DEF(5)   ,FXWOVS   ,  CINPUT
     4ISP(6)   ,ITPS     ,K(30)    ,KIN      ,LF       ,MR(6)    ,NCREW    ,  CINPUT
     5NENGS    ,NLISTO   ,NPASS    ,NWL      ,PCHAM    ,Q        ,RHOFU    ,  CINPUT
     6RHOFU2   ,RHOX     ,RHOX2    ,SBODY    ,TOL      ,TOVERC   ,TPRATO   ,  CINPUT
     7TYTAIL   ,VBODY    ,WGROSS                                             CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN    ,CTIP     ,GAL      ,GSPAN    ,  VOLCAL
     2HBODY    ,LBODY    ,RTOD     ,SFAIR    ,SFUTK    ,SHORZ    ,SOXTK    ,  VOLCAL
     3SPLAN    ,STPS(1)  ,SVERT    ,SWING    ,SXPOS    ,TDEL     ,TROOT    ,  VOLCAL
     4TTOT     ,TTOT2    ,TTOTAL   ,VBODYA   ,VBODY1   ,VBODY2   ,VCARGO   ,  VOLCAL
     5VCREW    ,VFUTK    ,VFUTK2   ,VINSTK   ,VLGBAY   ,VOTHER   ,VOXTK    ,  VOLCAL
     6VOXTK2   ,VPROP    ,VSTRUC                                             VOLCAL
      COMMON/WTCALC/     ABFSYS   ,WABFTK   ,WABFU    ,WABPR    ,WACRES   ,  WTCALC
     1WACS     ,WACSFO   ,WACSTK   ,WAERO    ,WAUXT    ,WBASIC   ,WBODY    ,  WTCALC
     2WBPUMP   ,WCARGO   ,WCOMM    ,WCONT    ,WCOVER   ,WDECAY   ,WDIST1   ,  WTCALC
     3WDIST2   ,WDOCK    ,WDPLOY   ,WDRANS   ,WDRY     ,WELCAD   ,WEMPTY   ,  WTCALC
     4WENGMT   ,WENGS    ,WENGS2   ,WFAIR    ,WFCONT   ,WFDCAY   ,WFROST   ,  WTCALC
     5WFU2(3)  ,WFUEL(6),WFUL     ,WFULOS   ,WFUNCT   ,WFUOX    ,WFURES   ,  WTCALC
     6WFUSYS   ,WFUTK    ,WFUTK2   ,WFUTOT   ,WFUTRP   ,WGASPR   ,WGNAV    ,  WTCALC
     7WHORZ    ,WHYCAD   ,WINFUT   ,WINOXT   ,WINSTK   ,WINST    ,WINSUL   ,  WTCALC
     8WJET(6)  ,WLANCH   ,WLG      ,WLOSS    ,WLRD     ,WNACEL   ,WODCAY   ,  WTCALC
     9WOIL     ,WOILRS   ,WORSUL   ,WOVERS   ,WOX(6)   ,WOX2(3)  ,WOXID    ,  WTCALC
     1WOXLOS   ,WOXRES   ,WOXSYS   ,WOXTK    ,WOXTK2   ,WOXTOT   ,WOXTRP   ,  WTCALC
     2WP       ,WPASS    ,WPAYL    ,WPERS    ,WPOWCD   ,WPOWER   ,WPOWFO   ,  WTCALC
     3WPOWRS   ,WPOWTK   ,WPPROV   ,WPREIG   ,WPROP    ,WPRSYS   ,WREFUL   ,  WTCALC
     4WRESID   ,WRESRV   ,WSEAL    ,WSECST   ,WSORCE   ,WSRTRP   ,WSTAB    ,  WTCALC
     5WSURF    ,WTABC    ,WTHRST   ,WTO      ,WTPS     ,WVERT    ,WWAIT(10),  WTCALC
     6WWET     ,WWING    ,WZROFU   ,WABTRP   ,WABRES   ,WMNOTP   ,WMNFTP   ,  CKOUT
     7WMNORS   ,WMNFRS   ,WACOTP   ,WACFTP   ,WPWOTP   ,WPWFTP   ,WGAS     ,  CKOUT
     8WABFUC   ,WACORS   ,WACFRS   ,WPWORS   ,WPWFRS                         CKOUT
      UWWING = 0.                                                       PROTHR
      UWVERT=0.0                                                        PROTHR
      UWHORZ=0.0                                                        PROTHR
      IF(SBODY.NE.0 0) UWBODY=(WBASIC+WSECST+WINFUT+WINOXT)/SBODY       PROTHR
      IF(SWING .NE. 0.)  UWWING = WWING / SWING                         PROTHR
      IF (SVERT.NE 0.) UWVERT = WVERT / SVERT                           PROTHR
      IF (SHORZ.NE.0.) UWHORZ = WHORZ / SHORZ                           PROTHR
      KILO = .0929                                                      PROTHR
      SFUTKK = KILO * SFUTK                                             PROTHR
      SOXTKK = KILO * SOXTK                                             PROTHR
      KSBODY = SBODY *KILO                                              PROTHR
      KSWING = SWING *KILO                                              PROTHR
      KSXPOS=KILO*SXPOS                                                 PROTHR
      KSPLAN =KILO*SPLAN                                                PROTHR
      KSVERT = SVERT *KILO                                              PROTHR
      KSHORZ = SHORZ *KILO                                              PROTHR
      KILO=4.8824                                                       PROTHR
      KUWWIN = KILO* UWWING                                             PROTHR
      KUWVER = KILO* UWVERT                                             PROTHR
      KUWHOR = KILO* UWHORZ                                             PROTHR
      KUWBOD = KILO* UWBODY                                             PROTHR
      KWOVER = WOVERS*KILO                                              PROTHR
      KILO = .0929**.5                                                  PROTHR
      KBBODY = KILO*BBODY                                               PROTHR
      KLROOT=KILO*CROOT                                                 PROTHR
      KLBODY = KILO*LBODY                                               PROTHR
      KCSPAN=CSPAN*KILO                                                 PROTHR
      KCROOT=CROOT*KILO                                                 PROTHR
      KHBODY = KILO*HBODY                                               PROTHR
      WRITE(6,40)                                                       PROTHR
   50 WRITE(6,100)                                                      PROTHR
  150 WRITE(6,200) SBODY     ,KSBODY                                    PROTHR
```

```
       WRITE(6,620) SFUTK,SFUTKK                                       PROTHR
       WRITE(6,625) SOXTK,SOXTKK                                       PROTHR
   650 WRITE(6,700)                                                    PROTHR
       WRITE(6,999) SWING,KSWING                                       PROTHR
       WRITE(6,3000) SXPOS,KSXPOS                                      PROTHR
       WRITE(6,9000)  SPLAN,KSPLAN                                     PROTHR
   950 WRITE(6,1000)SVERT      ,KSVERT                                 PROTHR
  1050 WRITE(6,1100)SHORZ      ,KSHORZ                                 PROTHR
  1250 WRITE(6,1300)                                                   PROTHR
  1350 WRITE(6,1400)UWWING,KUWWIN                                      PROTHR
  1450 WRITE(6,1500)UWVERT,KUWVER                                      PROTHR
  1550 WRITE(6,1600)UWHORZ,KUWHOR                                      PROTHR
  1650 IF(WINFUT.EQ.O..AND.WINOXT.EQ.O.)WRITE (6,1700) UWBODY, KUWBOD  PROTHR
   850 WRITE(6,910) WOVERS     ,KWOVER                                 PROTHR
 3250  WRITE(6,2700)                                                   PROTHR
       WRITE(6,2500)                                                   PROTHR
       WRITE(6,2000) CSPAN,KCSPAN                                      PROTHR
       WRITE(6,2100) CROOT,KCROOT                                      PROTHR
       WRITE(6,3100) TOVERC                                            PROTHR
       WRITE(6,2600)                                                   PROTHR
  3270 WRITE(6,2200)LBODY,KLBODY                                       PROTHR
       WRITE(6,2300) BBODY,KBBODY                                      PROTHR
       WRITE(6,2400) HBODY,KHBODY                                      PROTHR
       RETURN                                                          PROTHR
    40 FORMAT (/// 15X11HDESIGN DATA  )                                PROTHR
   100 FORMAT(1HO,12HWETTED AREAS   29X,10H  SQ. FT. ,10H    SQ.M.     PROTHR
      *)                                                               PROTHR
 200   FORMAT(1HO,40H    GROSS BODY (SBODY)                   2F10.2) PROTHR
   620 FORMAT (1H ,18HFUEL TANKS (SFUTK), 22X, 2F10.2)                 PROTHR
   625 FORMAT (1H ,22HOXIDIZER TANKS (SOXTK), 18X, 2F10.2)             PROTHR
   700 FORMAT(1HO,12HPLAN AREAS      29X,10H  SQ. FT. ,10H    SQ.M.  ) PROTHR
   999 FORMAT(1H ,40H    THEORETICAL WING AREA (SWING)        2F10.2) PROTHR
   910 FORMAT(1H ,40H    WING LOADING                         ,2F10.2) PROTHR
  1500 FORMAT(1H ,40H    VERTICAL SURFACES               2F10.2     ) PROTHR
  1100 FORMAT(1H ,40H    HORIZONTAL SURFACES (SHORZ)     2F10.2     ) PROTHR
 1300  FORMAT(1HO,12HUNIT WEIGHTS   28X,10H LB/SQ.FT.    ,10H  KG/SQ.M. ) PROTHR
  1400 FORMAT(1HO,40H    WING                            2F10.2     ) PROTHR
  1000 FORMAT(1H ,40H    VERTICAL SURFACES (SVERT)       2F10.2     ) PROTHR
  1600 FORMAT(1H ,40H    HORIZONTAL SURFACES             2F10.2     ) PROTHR
  1700 FORMAT(1H ,40H    BODY STRUCTURE (BASIC)          2F10.2     ) PROTHR
 2000  FORMAT(1H ,40H    STRUCTURAL SPAN                      2F10.2) PROTHR
 2100  FORMAT(1H ,40H    ROOT CHORD LENGTH                    2F10.2) PROTHR
  2200 FORMAT(1H ,40H    LENGTH (LBODY)                       2F10.2) PROTHR
  2300 FORMAT(1H ,40H    WIDTH  (BBODY)                       2F10.2) PROTHR
  2400 FORMAT(1H ,40H    HEIGHT  (HBODY)                      2F10.2) PROTHR
 2500  FORMAT(1H ,40H  WING                                         ) PROTHR
 2600  FORMAT(1H ,40H  BODY                                         ) PROTHR
 2700  FORMAT(1HO,16HDIMENSIONAL DATA,27X,10H FEET     ,10H METERS   ) PROTHR
  3000 FORMAT(1H ,40H    EXPOSED WING AREA (SXPOS)            2F10.2) PROTHR
  3100 FORMAT(1H ,30H    THICKNESS RATIO                      2F10.2) PROTHR
 9000  FORMAT(1H ,41H    PLANFORM, INCL. ELEVONS (SPLAN)               PROTHR
      *  F9.2,F10.2)                                                   PROTHR
       END                                                             PROTHR
```

SUBRØUTINE
PRWTSM

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
CFUEL		M	Mixture ratio	/CINPUT/(	306)	PRWTSM STORE WTSCH	M M M	CFUEL CFUEL CFUEL
I		W	Do loop counter	/PRWTSM/(*	)	PRWTSM	W	I
ISP		I	Specific impulse	/CINPUT/(	330)	PRWTSM STORE WTSCH WTVOL	I M I O	ISP ISP ISP ISP
J		W	Do loop counter	/PRWTSM/(*	)	PRWTSM	W	J
JUMP		M	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH PRINTW PRITVA PRWTSM WTSCH WTVOL	O I I M I M	JUMP JUMP JUMP JUMP JUMP JUMP
KDELTV		W	Velocity per arc (meters/sec)	/PRWTSM/(*	)	PRWTSM	W	KDELTV
KI		W	Constant = 6.0	/PRWTSM/(*	)	PRWTSM	W	KI
KILO		W	Pounds to kilogram conversion	/PRWTSM/(*	)	PRWTSM	W	KILO
KWABFU		W	Same as wabfu in mks units	/PRWTSM/(*	)	PRWTSM	W	KWABFU
KWFUEL		W	Same as wfuel(i) in mks units	/PRWTSM/(*	)	PRWTSM	W	KWFUEL
KWGROS		W	Gross weight (kilograms)	/PRWTSM/(*	)	PRWTSM	W	KWGROS
KWJET		W	Same as wjet(i) in mks units	/PRWTSM/(*	)	PRWTSM	W	KWJET
KWOX		W	Same as wox(i) in mks units	/PRWTSM/(*	)	PRWTSM	W	KWOX
KWPRIG		W	Pre-ignition loss (kg)	/PRWTSM/(*	)	PRWTSM	W	KWPRIG
KWWAIT		W	Same as wwait(i) in mks units	/PRWTSM/(*	)	PRWTSM	W	KWWAIT
MR		I	Mass ratio	/CINPUT/(	369)	PRWTSM SOLVE STORE WTSCH	I I M M	MR MR MR MR
OF		W	Oxidizer to fuel mixture ratio 1. Thrust build-up 2. Not used 3. Main impulse 4. Main impulse reserve 5. Secondary impulse 6. Not used	/PRWTSM/(*	)	PRWTSM	W	OF
PRWTSM		E	Subroutine to print summary data	/PRWTSM/($	)	PRWTSM WTVOL	E S	PRWTSM PRWTSM
WABFU		M	Weight of jp fuel	/WTCALC/(	3)	PRINTW PRWTSM TAMPER WTSCH	I M I M	WABFU WABFU WABFU WABFU
WAIT		W	Landing weight (lb)	/PRWTSM/(*	)	PRWTSM	W	WAIT
WAIT2		W	Landing weight (kg)	/PRWTSM/(*	)	PRWTSM	W	WAIT2
WENTRY		O	Entry weight	/EMS /(	3)	PRWTSM TAMPER	O I	WENTRY WENTRY
WETURN		O	Weight at return point	/EMS /(	2)	PRWTSM TAMPER	O I	WETURN WETURN
WFUEL		M	Fuel weight 1. Thrust build-up fuel 2. Not used 3. Main impulse fuel wt. 4. Main impulse fuel reserve 5. Secondary impulse fuel 6. Not used	/WTCALC/(	37)	PRWTSM WTSCH	M M	WFUEL WFUEL
WGROSS		I	Gross lift-off weight	/CINPUT/(	392)	PRINTW PRWTSM SOLVE STORE TAMPER WTSCH WTVOL	I I M M I M I	WGROSS WGROSS WGROSS WGROSS WGROSS WGROSS WGROSS
WJET		M	Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA PRWTSM STORE TAMPER WTSCH	I M O I M	WJET WJET WJET WJET WJET

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WLAND		O	Landing weight	/EMS /(	4)	PRWTSM	O	WLAND
						TAMPER	I	WLAND
WLOSS		I	In-flight weight loss	/WTCALC/(	70)	PRINTW	I	WLOSS
						PRWTSM	I	WLOSS
						WTSCH	O	WLOSS
WORBIT		O	Weight in orbit	/EMS /(	1)	PRWTSM	O	WORBIT
						TAMPER	I	WORBIT
WOX		M	Thrust build-up oxidizer 1. Thrust build-up oxidizer 2. Not used 3. Main impulse oxidizer 4. Main impulse oxidizer reserve 5. Secondary impulse oxidizer 6. Not used	/WTCALC/(	78)	PRWTSM	M	WOX
						WTSCH	M	WOX
WPREIG		M	Pre-ignition losses	/WTCALC/(	105)	PRINTW	I	WPREIG
						PRWTSM	M	WPREIG
						WTSCH	M	WPREIG
WWAIT		I	Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA	I	WWAIT
						PRWTSM	I	WWAIT
						STORE	M	WWAIT
						TAMPER	I	WWAIT
						WTSCH	M	WWAIT
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SOINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE PRWTSM                                                 PRWTSM
C                                                                       PRWTSM
C    PRWTSM - PRINT  WEIGHT SUMMARY                                     PRWTSM
C                                                                       PRWTSM
      REAL KWABFU                                                       PRWTSM
      REAL           KDELTV                                             PRWTSM
      REAL KWJET(6),KWFUEL(6),KWOX(6)                                   PRWTSM
     *       ,KWPRIG,KWGROS                                             PRWTSM
      REAL KILO,KWWAIT                                                  PRWTSM
      DIMENSION KDELTV(6)                                               PRWTSM
      DIMENSION DELTAV(6)                                               PRWTSM
      DIMENSION OF(6)                                                   PRWTSM
      DIMENSION KWWAIT(10)                                              PRWTSM
      COMMON / JUMPY / JUMP, WBIG, WBOO                                 PRWTSM
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFD  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
      COMMON/EMS/WORBIT,WETURN,WENTRY,WLAND                             PRWTSM
      WORBIT = 0.                                                       PRWTSM
      WLAND = 0.                                                        PRWTSM
      WENTRY = 0.                                                       PRWTSM
      WETURN = 0.                                                       PRWTSM
      KILO = 0.4535924                                                  PRWTSM
      KI= 6                                                             PRWTSM
      IF(WPREIG.NE.0.) WPREIG=-WPREIG                                   PRWTSM
    5 DO 111 I = 1, KI                                                  PRWTSM
      IF(CFUEL(I).EQ.0.0) CFUEL(I)=1.                                   PRWTSM
      OF(I) = (1-CFUEL(I)) / CFUEL(I)                                   PRWTSM
      IF(WJET(I).NE.0.) WJET(I)=-WJET(I)                                PRWTSM
      IF(WFUEL(I).NE.0.)WFUEL(I)=-WFUEL(I)                              PRWTSM
      IF(WOX(I).NE.0.)WOX(I)=-WOX(I)                                    PRWTSM
      KWJET(I)=WJET(I)*KILO                                             PRWTSM
      KWFUEL(I)=KILO*WFUEL(I)                                           PRWTSM
      KWOX(I)=KILO*WOX(I)                                               PRWTSM
      DELTAV(I)=32.174*ISP(I)*ALOG(MR(I))                               PRWTSM
      KDELTV(I)=DELTAV(I)/3.281                                         PRWTSM
  111 CONTINUE                                                          PRWTSM
      KWPRIG=KILO*WPREIG                                                PRWTSM
      KWGROS=KILO*WGROSS                                                PRWTSM
      DO 10 I=1,10                                                      PRWTSM
   10 KWWAIT(I) = KILO* WWAIT(I)                                        PRWTSM
      DO 20 J = 1,6                                                     PRWTSM
      IF(MR(J) .NE. 1.) GO TO 25                                        PRWTSM  25
```

```
   20 CONTINUE                                                              PRWTSM
      GO TO 50                                                              PRWTSM
   25 CONTINUE                                                              PRWTSM
      WRITE (6,664)                                                         PRWTSM
      WRITE(6,665)                                                          PRWTSM
      WRITE (6,666)(I,MR(I),OF(I),ISP(I),DELTAV(I),KDELTV(I), I=1,6)        PRWTSM
   50 WRITE(6,100)                                                          PRWTSM
      WRITE(6,240)                                                          PRWTSM
  150 WRITE(6,200) WGROSS,KWGROS                                            PRWTSM
      WRITE(6,205)WPREIG,KWPRIG                                             PRWTSM
      WRITE(6,207) WWAIT(1),KWWAIT(1)                                       PRWTSM
160   WRITE(6,210) WFUEL(1),KWFUEL(1)                                       PRWTSM
      WRITE(6,220) WOX  (1),KWOX  (1)                                       PRWTSM
      WRITE(6,230) WJET(1),KWJET(1)                                         PRWTSM
      WRITE(6,300) WWAIT(2),KWWAIT(2)                                       PRWTSM
      WENTRY = WWAIT(6)                                                     PRWTSM
      JUMP = JUMP + 1                                                       PRWTSM
      GO TO (260,280), JUMP                                                 PRWTSM
  260 WRITE (6,210) WFUEL(3), KWFUEL(3)                                     PRWTSM
      WRITE (6,220) WOX(3), KWOX(3)                                         PRWTSM
      WRITE(6,230) WJET(3),KWJET(3)                                         PRWTSM
      WRITE (6,401) WWAIT(4), KWWAIT(4)                                     PRWTSM
      WORBIT = WWAIT(4)                                                     PRWTSM
      WRITE(6,230) WJET(4),KWJET(4)                                         PRWTSM
      WRITE (6,500) WWAIT(5), KWWAIT(5)                                     PRWTSM
      WETURN = WWAIT(5)                                                     PRWTSM
      WRITE (6,210) WFUEL(5), KWFUEL(5)                                     PRWTSM
      WRITE (6,220) WOX(5), KWOX(5)                                         PRWTSM
      WRITE(6,230) WJET(5),KWJET(5)                                         PRWTSM
      WRITE (6,567) WWAIT(6), KWWAIT(6)                                     PRWTSM
      JUMP = JUMP - 1                                                       PRWTSM
      IF(JUMP.EQ 0   ) GO TO 765                                            PRWTSM
  280 WRITE (6,210) WFUEL(3), KWFUEL(3)                                     PRWTSM
      WRITE (6,220) WOX(3), KWOX(3)                                         PRWTSM
      WRITE(6,237) WJET(3),KWJET(3)                                         PRWTSM
      WRITE (6,1000) WWAIT(4), KWWAIT(4)                                    PRWTSM
      WETURN = WWAIT(4)                                                     PRWTSM
      WRITE (6,1005) WJET(4), KWJET(4)                                      PRWTSM
      WRITE (6,500) WWAIT(5)  , KWWAIT(5)                                   PRWTSM
      WRITE(6,1010) WJET(5),KWJET(5)                                        PRWTSM
      WRITE (6,1015)WWAIT(6), KWWAIT(6)                                     PRWTSM
      JUMP=JUMP-1                                                           PRWTSM
  765 KWABFU = WABFU * KILO                                                 PRWTSM
      WABFU = - WABFU                                                       PRWTSM
      KWABFU = - KWABFU                                                     PRWTSM
      WRITE (6,225) WABFU, KWABFU                                           PRWTSM
      WAIT  = WWAIT(7)                                                      PRWTSM
      WLAND =WAIT                                                           PRWTSM
      WABFU = - WABFU                                                       PRWTSM
      WAIT2 =KWWAIT(7)                                                      PRWTSM
      WRITE (6,600) WAIT, WAIT2                                             PRWTSM
      IF(WPREIG.NE.0.) WPREIG=-WPREIG                                       PRWTSM
      DO 120 I=1,KI                                                         PRWTSM
      IF(WJET(I).NE.0.) WJET(I)=-WJET(I)                                    PRWTSM
      IF(WFUEL(I).NE.0.)WFUEL(I)=-WFUEL(I)                                  PRWTSM
      IF(WOX(I).NE.0.)WOX(I)=-WOX(I)                                        PRWTSM
  120 CONTINUE                                                              PRWTSM
      IF(WJET(6).EQ.WLOSS) WJET(6) =0.                                      PRWTSM
      CALL PRITEQ                                                           PRWTSM
      RETURN                                                                PRWTSM
  100 FORMAT(// 15H WEIGHT SUMMARY  ,28X,10H  POUNDS   ,10HKILOGRAMS     )  PRWTSM
  200 FORMAT(1H0,40HMAXIMUM GROSS WEIGHT (WGROSS)              2F10.0)      PRWTSM
205   FORMAT(1H0,40H    PRE-IGNITION LOSSES                      2F10.0)    PRWTSM
207   FORMAT(1H0,40HMASS AT IGNITION                             2F10.0)    PRWTSM
  210 FORMAT(1H0,40H    FUEL                                     2F10.0)    PRWTSM
  220 FORMAT(1H ,40H    OXIDIZER                                 2F10.0)    PRWTSM
  225 FORMAT (1H0,40H    FLYBACK FUEL                            2F10.0)    PRWTSM
  230 FORMAT(1H ,40H    JETTISON                                 2F10.0)    PRWTSM
  237 FORMAT (1H ,40H    PAYLOAD ADDED/REMOVED                 2F10.0)      PRWTSM
  240 FORMAT(1H0,2H  )                                                      PRWTSM
  300 FORMAT(1H0,40HEND OF PRE-FLIGHT PHASE (WTO)                2F10.0)    PRWTSM
  401 FORMAT (1H0,40HWEIGHT IN ORBIT (WORBIT)                  2F10.0)      PRWTSM
  500 FORMAT(1H0,40HRETURN CONDITION (WETURN)                  2F10.0)      PRWTSM
```

```
      567 FORMAT(1H0,40HENTRY WEIGHT (WENTRY)                          2F10.0)  PRWTSM
      664 FORMAT (1H1)                                                          PRWTSM
      665 FORMAT(69X,31HTHEORETICAL VELOCITY INCREMENT    / 51X                 PRWTSM
         * 14HO/F RATIO  ISP,5X,28H FEET/SECOND   METERS/SECOND              ) PRWTSM
      666 FORMAT(1H0,11H MASS RATIO,20X,I2,1H=,F10.4,                           CKOUT
         *    F12.4,3XF6.1,2XF10.1,5XF10.1)                                     PRWTSM
      800 FORMAT(1H0,40HLANDING WEIGHT (WLAND)                    2F10.0  )     PRWTSM
     1000 FORMAT (1H0,40HINITIAL SEPARATION                       2F10.0)       PRWTSM
     1005 FORMAT (1H ,40H   THRUST DECAY AND RESIDUALS            2F10.0)       PRWTSM
     1010 FORMAT (1H ,40H   JETTISON AND EXPENDABLES              2F10.0)       PRWTSM
     1015 FORMAT (1H0,40HINITIAL FLYBACK                          2F10.0)       PRWTSM
          END                                                                   PRWTSM
```

SUBRØUTINE
RANGE

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
GAMS		W	Flyback flight path angle	/RANGE /(*	)	RANGE	W	GAMS
HA		W	Apogee altitude	/RANGE /(*	)	RANGE	W	HA
HAA		I	Apogee altitude	/RANGE /(*	)	RANGE	I	HAA
HAATBL		C	Table value of apogee alt	/RANGE /(*	)	RANGE	C	HAATBL
ISMD		W	Integer value of flybck	/RANGE /(*	)	RANGE	W	ISMD
PSI		W	Initial pitch attitude for range calculations	/RANGE /(*	)	RANGE	W	PSI
PSITBL		C	Table value of pitch attitude for range calculations	/RANGE /(*	)	RANGE	C	PSITBL
R		W	Range	/RANGE /(*	)	RANGE	W	R
RANGE		E	Subroutine to evaluate booster flyback range based on flybck	/RANGE /($	)	ITER8	S	RANGE
						RANGE	E	RANGE
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TBL2D		S	Two dimensional table look-up surboutine	/TBL2D /($	)	RANGE	S	TBL2D
						TBL2D	E	TBL2D
VA		W	Apogee velocity parameter used in flyback range determination	/RANGE /(*	)	RANGE	W	VA
VAA		I	Velocity parameter used in flyback range calculations	/RANGE /(*	)	RANGE	I	VAA
VAATBL		C	Staging velocity in range tables	/RANGE /(*	)	RANGE	C	VAATBL
VS		W	Staging velocity	/RANGE /(*	)	RANGE	W	VS
VSTBL		C	Table value of staging velocity	/RANGE /(*	)	RANGE	C	VSTBL

/1516

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	UN06.
						PAYOZ	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE RANGE                                                  RANGE
C         SUBROUTINE TO CALCULATE FLYBACK RANGE                         RANGE
C                                                                       RANGE
C                                                                       RANGE
      DIMENSION VSTBL(7), HSTBL(5), HANTBL(7,5), HAATBL(7,5),           RANGE
     2 VAATBL(7,5), VABTBL(7,5), RAATBL(7,5), RABTBL(7,5),              RANGE
     3 VATBL(7), HATBL(7), PSATBL(7,7),                                 RANGE
     4 HARTBL(7), PSITBL(4), HRATBL(7,4), HRBTBL(7,4)                   RANGE
C                                                                       RANGE
      REAL MUB, MUO, ISPB, ISPO, IOVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),     QP(14),   EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,     TLNG,                     SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,      DWEB,     DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,      BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,      OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,      IOVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,   NNB,        WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,      MUO,      VSTG,      WPO     -       SIZING
     *,JTYP,    BECO  ,    BSTG  ,   ORBI,    ITNBW  , ITNOW   ,          SIZING
     * SVOPSQ  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZD(19)                      UH
C                                                                       RANGE
C         PARAMETRIC RANGE DATA BASED ON                                RANGE
C              W/CL*S = 80.7 PSF                                        RANGE
C              L/D - MAX  AT CL-MAX=0.5                                 RANGE
C              CL-MAX = 0.723 AT ALPHA = 60 DEG                         RANGE
C              TERMINAL ALTITUDE = 50,000 FT                            RANGE
C              180-DEGREE MANEUVER                                      RANGE
C              MAXIMUM LOAD FACTOR = 4G                                 RANGE
C         DATA LIMITS                                                   RANGE
C              APOGEE ALTITUDE  = 300,000 FT                            RANGE
C              APOGEE VELOCITY  = 13,000 FPS                            RANGE
C              PATH ANGLE  = 14 DEGREES                                 RANGE
C                                                                       RANGE
      DATA HSTBL/ 150000., 175000., 200000., 225000., 250000./          RANGE
C                                                                       RANGE
      DATA VSTBL/ 7000., 8000., 9000., 10000., 11000., 12000., 13000./  RANGE
C                                                                       RANGE
      DATA HANTBL/ 2.175, 2.240, 2.320, 2.410, 2.505, 2.6075, 2.7125,   RANGE
     2 2.115, 2.145, 2.190, 2.250, 2.325, 2.405, 2.485,                 RANGE
     3 2.055, 2.073, 2.100, 2.125, 2.160, 2.200, 2.240,                 RANGE
     4 2.035, 2.038, 2.042, 2.050, 2.060, 2.080, 2.105,                 RANGE
     5 2.015, 2.020, 2.022, 2.025, 2.030, 2.032, 2.035/                 RANGE
C                                                                       RANGE
      DATA HAATBL/ 195., 200., 200., 197., 193., 187., 175.,            RANGE
     2 150., 183., 203., 215., 220., 220., 220.,                        RANGE
     3 200., 250., 303., 353., 405., 460., 510.,                        RANGE
     4 220., 290., 375., 470., 570., 670., 780.,                        RANGE
     5 235., 320., 415., 520., 640., 790., 880./                        RANGE
C                                                                       RANGE
      DATA VAATBL/ 3.65, 6.00, 8.25, 10.60, 12.90, 15.40, 18.25,        RANGE
     2 1.70, 2.30, 3.20, 4.40, 5.90, 7.55, 9.30,                        RANGE
     3 0.70, 1.15, 1.60, 2.35, 3.25, 4.45, 6.00,                        RANGE
     4 -1.00, -1.40, -1.00, 0.02, 1.70, 2.70, 2.70,                     RANGE
     5 -1.30, -1.80, -1.70, -1.25, -0.70, -0.45, -0.70/                 RANGE
C                                                                       RANGE
      DATA VABTBL/ -122.0,-166.0,-212.0,-260.0,-309.0,-360.0,-416.0,    RANGE
     2 -62.0, -70.0, -103.0, -133.0, -167.0, -202.0, -240.0,            RANGE
     3 -38.0, -47.0, -58.0, -73.0, -92.5, -119.0, -153.0,               RANGE
     4 -6.0, -10.0, -15.0, -22.0, -32.0, -46.0, -67.0,                  RANGE
     5 5.0, 2.5, 0., -2.5, -6.0, -12.5, -22.5/                          RANGE
C                                                                       RANGE
      DATA RAATBL/ 0.05, 0.065, 0.10, 0.18, 0.33, 0.385, 0.40,          RANGE
     2 0.02, 0.08, 0.16, 0.26, 0.39, 0.54, 0.73,                        RANGE
     3 0.07, 0.07, 0.07, 0.11, 0.15, 0.25, 0.52,                        RANGE
     4 -0.05, 0.01, 0.04, 0.055, 0.06, 0.098, 0.201,                    RANGE
     5 0.04, 0.03, 0.01, -0.018, -0.04, -0.05, 0.03/                    RANGE
C                                                                       RANGE
      DATA RABTBL/ 2.3, 2.8, 3.1, 2.8, 2.2, 2.5, 3.4,                   RANGE
     2 3.7, 4.1, 4.2, 4.9, 5.2, 5.55, 5.8,                              RANGE
     3 3.45, 4.85, 6.45, 8.3, 9.8, 10.9, 11.6,                          RANGE
     4 5.2, 5.2, 6.3, 9.0, 11.8, 14.25, 16.8,                           RANGE
     5 4.1, 5.9, 8.0, 10.5, 13.2, 16.1, 19.4/                           RANGE
```

```
      C                                                                    RANGE
            DATA VATBL/ 7000., 8000., 9000., 10000., 11000., 12000., 13000./   RANGE
      C                                                                    RANGE
            DATA HATBL/ 180000., 200000., 220000., 240000., 260000., 280000.,  RANGE
           1 300000./                                                      RANGE
      C                                                                    RANGE
            DATA PSATBL/ 90.0, 90.0, 89.0, 85.0, 83.0, 81.0, 79.5,         RANGE
           2 87.0, 80.5, 76.0, 73.5, 72.0, 71.0, 71.0,                     RANGE
           3 81.5, 66.5, 63.0, 62.0, 61.5, 61.5, 62.0,                     RANGE
           4 65.0, 58.5, 55.5, 54.0, 54.5, 56.0, 58.5,                     RANGE
           5 35.0, 32.5, 32.0, 33.0, 35.0, 38.0, 41.5,                     RANGE
           6 17.0, 16.0, 17.0, 19.0, 22.0, 25.5, 30.0,                     RANGE
           7  0.0,  0.0,  1.5,  4.0,  8.0, 13.0, 19.0/                     RANGE
      C                                                                    RANGE
            DATA PSITBL/ 0., 45., 60., 75./                                RANGE
      C                                                                    RANGE
            DATA HARTBL/150000., 175000., 200000., 225000.,                RANGE
           1 250000., 275000., 300000./                                    RANGE
      C                                                                    RANGE
            DATA HRATBL/ 7*16.0E-07,                                       RANGE
           1 7 985E-07, 7.950E-07, 7.915E-07, 7.955E-07,                   RANGE
           1 7.994E-07, 8.000E-07, 8.040E-07,                              RANGE
           2 3 204E-07, 4.100E-07, 4.636E-07, 3.900E-07,                   RANGE
           2 5.176E-07, 5.350E-07, 5.541E-07,                              RANGE
           3 0.297E-07, 1.150E-07, 1.969E-07, 2.600E-07,                   RANGE
           3 3.131E-07, 3.550E-07, 3.958E-07/                              RANGE
      C                                                                    RANGE
            DATA HRBTBL/ 0., 0.250E-02, 0.475E-02, 0.650E-02,              RANGE
           A 0.800E-02, 0.950E-02, 1.100E-02,                              RANGE
           1 0.577E-02, 0.600E-02, 0.992E-02, 1.150E-02,                   RANGE
           1 1.276E-02, 1.400E-02, 1.512E-02,                              RANGE
           2 0.830E-02, 1.000E-02, 1.152E-02, 1.275E-02,                   RANGE
           2  1.425E-02, 1 525E-02, 1.652E-02,                             RANGE
           3 0.931E-02, 1.120E-02, 1.271E-02, 1.400E-02,                   RANGE
           3 1.522E-02, 1.630E-02, 1.726E-02/                              RANGE
      C                                                                    RANGE
      C     BANK ANGLE DURING ENTRY = PSI = SQ(10,2)                       RANGE
            SQ(10,2) = 0.                                                  RANGE
      C                                                                    RANGE
      C         TEST FLYBACK OPTION FLAG                                   RANGE
      C                                                                    RANGE
            ISND = IFIX(SQ(19,5)+.1)                                       RANGE
            GO TO (100,200,300,400,500), ISND                              RANGE
      C                                                                    RANGE
        100 CONTINUE                                                       RANGE
      C     DETAILED FLYBACK RANGE CALCULATIONS                            RANGE
            VS = SV(8)                                                     RANGE
            HS = SV(9)                                                     RANGE
      C         CHECK STAGING ALTITUDE                                     RANGE
            IF(HS.GT 250000.) WRITE(6,1000)                                RANGE
            IF(HS GT.250000.) HS = 250000.                                 RANGE
            IF( SV(10) .LE. 0. ) GAMS= 0.0001                              FINI
            GAMS = SV(10)                                                  RANGE
      C         CHECK STAGING PATH ANGLE                                   RANGE
            IF(GAMS.GT.14.)  WRITE(6,2000)                                 RANGE
            IF(GAMS.GT.14.) GAMS = 14.                                     RANGE
            DRS = SW(10)                                                   RANGE
      C                                                                    RANGE
      C         CALCULATE APOGEE ALTITUDE                                  RANGE
      C                                                                    RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,HANTBL,HAN)                      RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,HAATBL,HAA)                      RANGE
            HA = HAA*GAMS**HAN + HS                                        RANGE
      C                                                                    RANGE
      C         CALCULATE APOGEE VELOCITY                                  RANGE
      C                                                                    RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,VAATBL,VAA)                      RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,VABTBL,VAB)                      RANGE
            VA = VAA*GAMS**2. + VAB*GAMS + VS                              RANGE
      C                                                                    RANGE
      C         CALCULATE RANGE TO APOGEE                                  RANGE
      C                                                                    RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,RAATBL,RAA)                      RANGE
            CALL TBL2D( VS,HS,VSTBL,HSTBL,RABTBL,RAB)                      RANGE
            RA = RAA*GAMS**2. + RAB*GAMS                                   RANGE
```

100 200 300 400 500

20 OCT 72 6 01-46

```
C                                                                    RANGE
C         TEST APOGEE ALTITUDE                                       RANGE
C                                                                    RANGE
      IF(HA.GT.300000.) WRITE(6,3000)                                RANGE
      IF(HA.GT.300000.) HA = 300000.                                 RANGE
C                                                                    RANGE
C         RETREIVE BANK ANGLE REQUIRED AND TEST                      RANGE
C                                                                    RANGE
      CALL TBL2D(VA,HA,VATBL,HATBL,PSATBL,PSI)                       RANGE
      IF(PSI.GE.75.0) PSI = 75.0                                     RANGE
      SQ(10,2) = PSI                                                 RANGE
C                                                                    RANGE
C         CALCULATE RANGE                                            RANGE
C                                                                    RANGE
      CALL TBL2D(HA,PSI,HARTBL,PSITBL,HRATBL,HRA)                    RANGE
      CALL TBL2D(HA,PSI,HARTBL,PSITBL,HRBTBL,HRB)                    RANGE
      RR = HRA*VA**2. + HRB*VA                                       RANGE
C                                                                    RANGE
      R = 60.*DRS + RA + RR + SQ(10,3)                               RANGE
C                                                                    RANGE
C     SQ(10,3) IS A FLYBACK ADDITIVE RANGE FACTOR TO ACCOUNT FOR THE RANGE
C     FINITE TIME REQUIRED TO ACCOMPLISH VARIOUS REENTRY MANEUVERS.  RANGE
C     DRNG IS THE NUMONIC INPUT IN DATA2.                            RANGE
C                                                                    RANGE
C         IF VARABLE GEOMETRY CONFIGURATION GO TO 10                 RANGE
C                                                                    RANGE
      IF(SW(9).NE.1.) GO TO 10                                       RANGE
C                                                                    RANGE
      SW(15) = R                                                     RANGE
C                                                                    RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
C     CORRECT FG RANGE PREDICTION FOR VG CONFIGURATION               RANGE
   10 SW(15) = R*(0.7230/SW(9))**0.25                                RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
  200 CONTINUE                                                       RANGE
C     FLYBACK RANGE = FUNCTION OF STAGING Q ONLY                     RANGE
C                                                                    RANGE
C     SPECIAL FLYBACK RANGE COMPUTATION FOR MSC                      RANGE
C     VQ=  LOG(10) Q STAGE                                           RANGE
C                                                                    RANGE
      VQ = ALOG10(SV(12))                                            CKOUT
      RNG = -55.915 * VQ + 430.0                                     RANGE
C                                                                    RANGE
C     SQ(10,3) IS A FLYBACK ADDITIVE RANGE FACTOR TO ACCOUNT FOR THE RANGE
C     FINITE TIME REQUIRED TO ACCOMPLISH VARIOUS REENTRY MANEUVERS.  RANGE
C     DRNG IS THE MNEMONIC INPUT IN DATA2.                           RANGE
C     SW(15) = FLYBACK RANGE                                         RANGE
      SW(15) = RNG  +  SQ(10,3)                                      RANGE
C                                                                    RANGE
C                                                                    RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
  300 CONTINUE                                                       RANGE
C     FLYBACK RANGE = CONSTANT = SQ(10,3) INPUT IN DATA2             RANGE
      SW(15) = SQ(10,3)                                              RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
  400 CONTINUE                                                       RANGE
C     FLYBACK RANGE = IIP                                            RANGE
C     SW(10) = CENTRAL ANGLE FROM LIFT OFF TO STAGING                RANGE
C     SQ(12,1) = IIP CENTRAL ANGLE FROM STAGING TO IMPACT = Z(89)    RANGE
C     SQ(10,3) = ADDITIVE RANGE FACTOR (DATA2 INPUT)                 RANGE
      SW(15) = 60.*(SQ(12,1) + SW(10)) + SQ(10,3)                    RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
  500 CONTINUE                                                       RANGE
C  ****  FLYBACK TRAJECTORY TO BE NUMERICALLY INTEGRATED  ****       RANGE
      RETURN                                                         RANGE
C                                                                    RANGE
 1000 FORMAT(74H STAGING ALTITUDE GREATER THAN 250000 FT - FLYBACK RANGE RANGE
     1 EQUATIONS INVALID)                                            RANGE
```

```
 2000 FORMAT(68H STAGING GAMMA GREATER THAN 14 DEG - FLYBACK RANGE EQUAT RANGE
     1IONS INVALID)                                                       RANGE
 3000 FORMAT(73H APOGEE ALTITUDE GREATER THAN 300000 FT - FLYBACK RANGE  RANGE
     1EQUATIONS INVALID)                                                  RANGE
      END                                                                 RANGE
```

SUBRØUTINE SETO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
A		O	Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH FRENCH SET0 STORE WTSCH	I M O M I	A ANENGS A ANENGS ANENGS
C		O	Airbreathing fuel system weight used by set0 to set common to zero	/WTCALC/(	1)	SET0 WTSCH	O M	C ABFSYS
D		O	Body width used by set0 to set common to zero	/VOLCAL/(	1)	PROTHR SET0 WTSCH	I O M	BBODY D BBODY
I		W	Do loop counter	/SET0 /(*	)	SET0	W	I

```
      SUBROUTINE SETO                                                   SETO
C********************************************************************** SETO
C     INITIALIZES VEHICLE DATA TO ZERO                                  SETO
C********************************************************************** SETO
      COMMON/CINPUT/A(392)                                              SETO
      COMMON/VOLCAL/D(37)                                               SETO
      COMMON/WTCALC/C(134)                                              SETO
      DO 10 I=1,392                                                     SETO
   10 A(I)=0                                                            SETO
      DO 30 I=1,134                                                     SETO
   30 C(I)=0                                                            SETO
      DO 40 I=1,37                                                      SETO
   40 D(I)=0                                                            SETO
      RETURN                                                            SETO
      END                                                               SETO
```

SUBRØUTINE
SIZEMR

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SE		I	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		M	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB27		I	Stored booster value of Isp(I)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TO27		I	Stored orbiter value of Isp(I)	/ORBINX/(	41)	SIZEMR	I	TO27
						SSSP	I	TO27
						STORE	M	TO27
						SUMOUT	I	TO27
						VEHDF	M	TO27
						WTVOL	I	TO27

```
      SUBROUTINE SIZEMR                                                    SIZEMR
C                                                                          SIZEMR
C         SUBROUTINE TO EVALUATE MAIN IMPULSE MASS RATIOS                  SIZEMR
C                                                                          SIZEMR
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                              SIZING
      COMMON /SIZING/                                                      SIZING
C          PHASE II SIZING PARAMERERS                                      SIZING
     *TZ,      VV(3),     QP(14),   EROR,     PZ(5),     VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),   TLAT,     TLNG,                        SIZING
C         PHASE I SIZING PARAMERERS                                        SIZING
     *WBO,     WLOO,      DWEB,     DWEO,     TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,     BK2,       BK3,      BK4,      ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,     OK2,       OK3,      OK4,      PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,       WFO,      IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,     TVACB,   NNB,        WEO,      WEB,       WO,       WLO,    SIZING
     *DVO,      DVB,       MUB,      MUO,      VSTG,      WPO              SIZING
     *,JTYP,    BECO   ,  BSTG      ORBI      ITNBW   ,  ITNOW   ,         SIZING
     * SVDPSQ  , SVDCON  ,IHUNT   ,IOPSTG   ,ISZD(19)                      UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),          ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                       CKOUT
      COMMON/ORBINX/                                                       ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,       ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27,   ORBINX
     3 TO28,TO29,TO30,TO31,TO32,TO33,TO34,TO35,TO36,TO37,TO38,TO39,TO40,   ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53,   ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                            CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,        UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,        UH
     8TO84                                                                 UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),          ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                       CKOUT
      COMMON/ORBINY/                                                       ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,       ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27,   ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40,   ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53,   ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                  UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,        UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,         UH
     8TB83,TB84                                                            UH
C                                                                          SIZEMR
C         CALCULATE LAST STAGE MASS RATIO                                  SIZEMR
C                                                                          SIZEMR
      SV(6) = SV(7)/SV(4)                                                  CKOUT
C                                                                          SIZEMR
C         CALCULATE BOOSTER MASS RATIO                                     SIZEMR
C                                                                          SIZEMR
      SV(28) = VV(1)/QP(9)                                                 CKOUT
C                                                                          SIZEMR
C         TEST PARALLEL BURN FLAG                                          SIZEMR
C                                                                          SIZEMR
      IF( SE(2).NE.1.) GO TO 1                                             SIZEMR
C                                                                          SIZEMR
C         CALCULATE EFFECTIVE FIRST STAGE MASS RATIO                       SIZEMR
C                                                                          SIZEMR
      ORBDV     = 32.174 * TO27(3) * ALOG(SV(6))                           CKOUT
      EFFMR     = EXP((SV(3)- ORBDV)/32.174/QP(13))                        SIZEMR
C                                                                          SIZEMR
C         DETERMINE BOOSTER SIZING MASS RATIO                              SIZEMR
C                                                                          SIZEMR
      TBPAR     = VV(1)/EFFMR *(EFFMR -1.)* QP(13) / QP(1)                 SIZEMR
      TBPARO = TBPAR * TO27(3)/ QP(2)                                      CKOUT
      SV(28) = (VV(1) - WPPARO)/QP(9)                                      CKOUT
      GO TO 99                                                             SIZEMR
C                                                                          SIZEMR
C         CHECK CROSS FEED FLAG                                            SIZEMR
C                                                                          SIZEMR
    1 IF( SE(10).EQ.1.)  GO TO 2                                           SIZEMR
C                                                                          SIZEMR
C         CORRECT ORBITER MASS RATIO                                       SIZEMR
C                                                                          SIZEMR
      SV(6) = (SV(7) + SE(11))/ SV(4)                                      CKOUT
C                                                                          SIZEMR
C         CALCULATE BOOSTER MASS RATIO                                     SIZEMR
C                                                                          SIZEMR
```

```
      SV(28) = (VV(1) - SE(11)) / QP(9)                                 CKOUT
C                                                                       SIZEMR
C          CHECK SOLID MOTOR FLAG                                       SIZEMR
C                                                                       SIZEMR
    2 IF( SQ(20,1).LE.0.)  GO TO 99                                     SIZEMR
C                                                                       SIZEMR
C          CALCULATE BOOSTER MASS RATIO                                 SIZEMR
C                                                                       SIZEMR
       SV(28) = (VV(1)- SQ(21,5))/ QP(9)                                CKOUT
C                                                                       SIZEMR
C          CALCULATE VELOCITIES                                         SIZEMR
C                                                                       SIZEMR
   99 CONTINUE                                                          SIZEMR
      SQ(14,1) =  32.174 * TB27(3) * ALOG(SV(28))                       CKOUT
      SQ(14,3) =  32.174 * TO27(3) * ALOG(SV(6))                        CKOUT
      SQ(14,2) = SV(3)                                                  CKOUT
      SV(2)=SV(3)                                                       FINI
  999 RETURN                                                            SIZEMR
      END                                                               SIZEMR
```

SUBRØUTINE
SØLVE

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
MR		I	Mass ratio	/CINPUT/(	369)	PRWTSM	I	MR
						SOLVE	I	MR
						STORE	M	MR
						WTSCH	M	MR
SOLVE		E	Subroutine to drive weight and volume calculations (wtsch) to convergence - an inner loop driver	/SOLVE /($	)	SOLVE	E	SOLVE
						WTVOL	S	SOLVE
TOL		M	Gross weight iteration tolerance	/CINPUT/(	387)	SOLVE	M	TOL
VBODY		M	Total body volume	/CINPUT/(	391)	PRINTV	M	VBODY
						SOLVE	M	VBODY
						STORE	M	VBODY
						TAMPER	I	VBODY
						WTSCH	M	VBODY
						WTVOL	I	VBODY
V1		W	Dummy value of vbody in solve routin	/SOLVE /(*	)	SOLVE	W	V1
V2		W	Dummy value of vbody in solve routin	/SOLVE /(*	)	SOLVE	W	V2
WGROSS		M	Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
						PRWTSM	I	WGROSS
						SOLVE	M	WGROSS
						STORE	M	WGROSS
						TAMPER	I	WGROSS
						WTSCH	M	WGROSS
						WTVOL	I	WGROSS
WTSCH		S	Subroutine to calculate weight and volume of both stages	/WTSCH /($	)	SOLVE	S	WTSCH
						WTSCH	E	WTSCH
						WTVOL	S	WTSCH
W1		W	Dummy value of wgross in solve loop	/SOLVE /(*	)	SOLVE	W	W1
W2		W	Dummy value of wgross in solve loop	/SOLVE /(*	)	SOLVE	W	W2
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE SOLVE                                                  SOLVE
C   THIS VERSION OF SOLVE DRIVES SUBROUTINE WTSCH TO CLOSE ON WEIGHT    SOLVE
C  USING A NEWTON/RAPHSON ITERATION SCHEME                              SOLVE
C                                                                       SOLVE
      REAL MIN,MAX                                                      SOLVE
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C           PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,      VV(3),    QP(14),   EROR,     PZ(5),    VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5), SE(11),   TLAT,     TLNG,                      SIZING
C          PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,     WLOD,     DWEB,     DWEO,     TOLWT,    WPB,      TWRAT2, SIZING
     *BK1,     BK2,      BK3,      BK4,      ISIZE,    TRAFLG,   TWRATO, SIZING
     *OK1,     OK2,      OK3,      OK4,      PRFLG,    IPASS,    IPSMAX, SIZING
     *AEXIT,   TVACO,    NO,       WFO,      IDVEL,    ISPO,     ISPB,   SIZING
     *XPL,     TVACB,    NNB,      WEO,      WEB,      WO,       WLO,    SIZING
     *DVO,     DVB,      MUB,      MUO,      VSTG,     WPO               SIZING
     *,JTYP,   BECO  ,   BSTG  ,   ORBI ,    ITNBW , ITNOW ,             SIZING
     * SVOPSQ  , SVOCON  ,IHUNT  ,IOPSTG  ,ISZD(19)                     UH
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
C                                                                       SOLVE
      IF(SW(13).GT.2.5) WRITE(6,1001)                                   SOLVE
C                                                                       SOLVE
      WK=MR(3)                                                          SOLVE
      IF(TOL.LT.1..OR.TOL.GT.100.) TOL = 5.                             SOLVE
      ITER =0                                                           SOLVE
      MAX=2.5*WGROSS                                                    SOLVE
      MIN=0.4*WGROSS                                                    SOLVE
C                                                                       SOLVE
      V1=VBODY                                                          SOLVE
      W1 = WGROSS                                                       SOLVE
      CALL WTSCH                                                        SOLVE
 5005     CONTINUE                                                      SOLVE
      DW1=WGROSS-W1                                                     SOLVE
      DV1=VBODY -V1                                                     SOLVE
C                                                                       SOLVE
   10 CONTINUE                                                          SOLVE
      W2=WGROSS+WK*DW1                                                  SOLVE
      R=W2/WGROSS                                                       SOLVE
      V2=VBODY*R                                                        SOLVE
```

```
      VBODY=V2                                                          SOLVE
      WGROSS=W2                                                         SOLVE
      ITER=ITER+1                                                       SOLVE
      CALL WTSCH                                                        SOLVE
      DW2=WGROSS-W2                                                     SOLVE
      DV2=VBODY -V2                                                     SOLVE
      IF(W2.GT.MAX.OR.W2.LT.MIN) GO TO 120                              SOLVE
      IF(ABS(DW2).LT.TOL.AND.ABS(DV2).LT.5.) GO TO 100                  SOLVE
      IF(ITER GT.15)GO TO 110                                           SOLVE
      WK=(W2-W1)/(DW1-DW2) - 1.                                         SOLVE
      IF(WK.GT.5.) WK=5.                                                SOLVE
      IF(WK.LT.2.) WK=2.                                                SOLVE
      W1 = W2                                                           SOLVE
      DW1 = DW2                                                         SOLVE
      V1 = V2                                                           SOLVE
      DV1 = DV2                                                         SOLVE
      IF(SW(13).GT.2.5) WRITE(6,1000)  W2,DW2, V2,DV2, WK,ITER          SOLVE
      IF(0) 5005,10,5005                                                SOLVE
C     THIS CARD NECESSARY TO PROGRAM AROUND COMPILER                    SOLVE
C     OPTIMIZATION BUG ON UNIVAC 1108 EXEC II.                          SOLVE
C                                                                       SOLVE
C                                                                       SOLVE
  110 CONTINUE                                                          SOLVE
  100 CONTINUE                                                          SOLVE
      IF(SW(13) EQ.2.0) WRITE(6,1001)                                   SOLVE
      IF(SW(13).GT.1.5) WRITE(6,1000)  W2,DW2, V2,DV2, WK,ITER          SOLVE
      RETURN                                                            SOLVE
  120 CONTINUE                                                          SOLVE
      WRITE(6,1002)                                                     SOLVE
 1002 FORMAT(10X,30H ***** BLOWUP IN SOLVE *****    //)                 SOLVE
      WRITE(6,1001)                                                     SOLVE
                           WRITE(6,1000)  W2,DW2, V2,DV2, WK,ITER       SOLVE
      RETURN                                                            SOLVE
 1000 FORMAT( 5X,2( F12.2,F12.4),F12.3,I4 )                             SOLVE
 1001 FORMAT(/9X,2HW2 10X,3HDW2 8X, 2HV2 10X, 3HDV2 10X,                SOLVE
     1 2HWK 8X,4HITER  )                                                SOLVE
C                                                                       SOLVE
      END                                                               SOLVE
```

SUBRØUTINE STØRE

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	SUBROUTINE USAGE CODE	VAR
ANENGS		M	Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH	I	A
						FRENCH	M	ANENGS
						SETO	O	A
						STORE	M	ANENGS
						WTSCH	I	ANENGS
ANTANK		M	Number of air breathing fuel tanks	/CINPUT/(	2)	STORE	M	ANTANK
						WTSCH	I	ANTANK
ASRATO		M	Wing aspect ratio	/CINPUT/(	3)	STORE	M	ASRATO
						WTSCH	I	ASRATO
ASWEEP		M	Wing leading edge sweep angle	/CINPUT/(	4)	STORE	M	ASWEEP
						WTSCH	I	ASWEEP
C		M	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
						PRITEQ	I	C
						PRITVA	I	C
						STORE	M	C
						WTSCH	I	C
						WTVOL	O	C
CBBODY		M	Body width coeff.	/CINPUT/(	305)	PRITVA	I	CBBODY
						STORE	M	CBBODY
						WTSCH	I	CBBODY
CFUEL		M	Mixture ratio	/CINPUT/(	306)	PRWTSM	M	CFUEL
						STORE	M	CFUEL
						WTSCH	M	CFUEL
CHBODY		M	Body height or coeff	/CINPUT/(	312)	PRITVA	I	CHBODY
						STORE	M	CHBODY
						WTSCH	I	CHBODY
CLBODY		M	Body length or coeff	/CINPUT/(	313)	PRITVA	I	CLBODY
						STORE	M	CLBODY
						WTSCH	I	CLBODY
CSBODY		M	Total body wetted area or coeff	/CINPUT/(	314)	PRITVA	I	CSBODY
						STORE	M	CSBODY
						WTSCH	I	CSBODY
CSFAIR		M	Fairing planform area or coeff	/CINPUT/(	315)	PRITVA	I	CSFAIR
						STORE	M	CSFAIR
						WTSCH	I	CSFAIR
CSFUTK		M	Fuel tank surface area or coeff	/CINPUT/(	316)	PRITVA	I	CSFUTK
						STORE	M	CSFUTK
						WTSCH	I	CSFUTK
CSHORZ		M	Horizontal stabalizer planform area	/CINPUT/(	317)	PRITVA	I	CSHORZ
						STORE	M	CSHORZ
						WTSCH	I	CSHORZ
CSOXTK		M	Oxidizer tank surface area coeff	/CINPUT/(	318)	PRITVA	I	CSOXTK
						STORE	M	CSOXTK
						WTSCH	I	CSOXTK
CSPLAN		M	Body planform area or coeff	/CINPUT/(	319)	PRITVA	I	CSPLAN
						STORE	M	CSPLAN
						WTSCH	I	CSPLAN
CSVERT		M	Vertical fin planform area or coeff	/CINPUT/(	320)	PRITVA	I	CSVERT
						STORE	M	CSVERT
						WTSCH	I	CSVERT
CSWING		M	Wing planform area	/CINPUT/(	321)	STORE	M	CSWING
CTHRST		M	Vac. Thrust-to-weight ratio	/CINPUT/(	322)	PRITVA	I	CTHRST
						STORE	M	CTHRST
						WTSCH	I	CTHRST
						WTVOL	M	CTHRST
CTHST2		M	Secondary propulsion t/w	/CINPUT/(	323)	PRITVA	I	CTHST2
						STORE	M	CTHST2
						WTSCH	I	CTHST2
FXWOVS		M	Fixed wing loading	/CINPUT/(	329)	STORE	M	FXWOVS
						WTSCH	I	FXWOVS
I		W	Do loop counter	/STORE /(*	)	STORE	W	I

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
ISP		M	Specific impulse	/CINPUT/(	330)	PRWTSM STORE WTSCH WTVOL	I M I O	ISP ISP ISP ISP
ITPS		M	Thermo protection flag	/CINPUT/(	336)	FRENCH STORE WTSCH	O M M	ITPS ITPS ITPS
LF		M	Ultimate load factor 1. Thrust buildup 2. Not used 3. Main impulse mass ratio 4. Main impulse reserve 5. Secondary impulse mass ratio 6. Not used	/CINPUT/(	368)	STORE WTSCH	M I	LF LF
MR		M	Mass ratio	/CINPUT/(	369)	PRWTSM SOLVE STORE WTSCH	I I M M	MR MR MR MR
NCREW		M	Number of crew members	/CINPUT/(	375)	PRITVA STORE WTSCH	I M I	NCREW NCREW NCREW
NENGS		M	Total number engines per stage	/CINPUT/(	376)	PRITVA STORE WTSCH	I M I	NENGS NENGS NENGS
NLISTO		M	Namelist output flag	/CINPUT/(	377)	STORE WTSCH	M I	NLISTO NLISTO
NPASS		M	Number of passengers	/CINPUT/(	378)	STORE WTSCH	M I	NPASS NPASS
NWL		M	Wing loading flag	/CINPUT/(	379)	FRENCH STORE WTSCH	O M M	NWL NWL NWL
PCHAM		M	Main rocket engine chamber pressure	/CINPUT/(	380)	STORE WTSCH	M I	PCHAM PCHAM
RHOFU		M	Fuel density	/CINPUT/(	382)	PRITVA STORE WTSCH	I M I	RHOFU RHOFU RHOFU
RHOFU2		M	Secondary fuel desnity	/CINPUT/(	383)	PRITVA STORE WTSCH	I M I	RHOFU2 RHOFU2 RHOFU2
RHOX		M	Oxidizer density	/CINPUT/(	384)	PRITVA STORE WTSCH	I M I	RHOX RHOX RHOX
RHOX2		M	Secondary oxidizer density	/CINPUT/(	385)	PRITVA STORE WTSCH	I M I	RHOX2 RHOX2 RHOX2
SCB		M	Working name for input c-array booster scaling coefficients	/ORBINY/(	144)	FLYBKP STORE SUMOUT TAMPER THRUST VEHOF WTVOL	M M I I M I M	SCB SCB SCB SCB SCB SCB SCB
SKB		M	Working name for input k-array booster volume scaling coeff	/ORBINY/(	114)	STORE	M	SKB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SQRT		F	Square root function	/SQRT /($	)	ANLATM	F	SQRT
						CRASH	F	SQRT
						OCTOE	F	SQRT
						DER3A	F	SQRT
						ENVPRM	F	SQRT
						HUNT	F	SQRT
						MODELA	F	SQRT
						MODELB	F	SQRT
						OPWELL	F	SQRT
						OUT	F	SQRT
						PAT63	F	SQRT
						PAY02	F	SQRT
						PDBC	F	SQRT
						PDY3A	F	SQRT
						STORE	F	SQRT
						SYMVRT	F	SQRT
						WTSCH	F	SQRT
STORE		E	Subroutine to store vehicle data in internal format and wtvol format	/STORE /($	)	FRENCH	S	STORE
						STORE	E	STORE
SWING		M	Gross wing area	/VOLCAL/(	17)	PROTHR	I	SWING
						STORE	M	SWING
						WTSCH	M	SWING
TB10		M	Stored booster value of csfair	/ORBINY/(	15)	STORE	M	TB10
TB11		M	Stored booster value of csfutk	/ORBINY/(	16)	STORE	M	TB11
TB12		M	Stored booster value of csoxtk	/ORBINY/(	17)	STORE	M	TB12
TB13		M	Stored booster value of cshorz	/ORBINY/(	18)	STORE	M	TB13
TB15		M	Stored booster value of csplan	/ORBINY/(	20)	STORE	M	TB15
TB16		M	Stored booster value of csvert	/ORBINY/(	21)	STORE	M	TB16
TB17		M	Stored booster value of cswing	/ORBINY/(	22)	STORE	M	TB17
TB18		M	Stored booster value of cthrst	/ORBINY/(	23)	STORE	M	TB18
						WTVOL	O	TB18
TB19		M	Stored booster value of cthst2	/ORBINY/(	24)	STORE	M	TB19
TB2		M	Stored booster value of cbbody	/ORBINY/(	2)	STORE	M	TB2
TB27		M	Stored booster value of isp(i)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		M	Stored booster value of mr(i)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TB35		M	Stored booster value of ncrew	/ORBINY/(	59)	STORE	M	TB35
TB36		M	Stored booster value of nengs	/ORBINY/(	60)	STORE	M	TB36
						SUMOUT	I	TB36
						TAMPER	I	TB36
						THRUST	I	TB36
						WTVOL	I	TB36
TB38		M	Stored booster value of npass	/ORBINY/(	62)	STORE	M	TB38
TB4		M	Stored booster value of lf	/ORBINY/(	4)	STORE	M	TB4
TB41		M	Stored booster value of rhofu	/ORBINY/(	65)	STORE	M	TB41
TB42		M	Stored booster value of rhofu2	/ORBINY/(	66)	STORE	M	TB42
TB43		M	Stored booster value of rhox	/ORBINY/(	67)	STORE	M	TB43
TB44		M	Stored booster value of rhox2	/ORBINY/(	68)	STORE	M	TB44

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
TB45		M	Stored booster value of swing	/ORBINY/(	69)	STORE SUMOUT TAMPER	M I I	TB45 TB45 TB45
TB47		M	Stored booster value of toverc	/ORBINY/(	71)	STORE	M	TB47
TB5		M	Stored booster value of chbody	/ORBINY/(	10)	STORE	M	TB5
TB51		M	Stored booster value of vbody	/ORBINY/(	102)	STORE WTVOL	M O	TB51 TB51
TB52		O	Stored booster value of vfutk	/ORBINY/(	103)	STORE	O	TB52
TB53		M	Stored booster value of vfutk2	/ORBINY/(	104)	STORE	M	TB53
TB54		O	Stored booster value of voxtk	/ORBINY/(	105)	STORE	O	TB54
TB55		M	Stored booster value of voxtk2	/ORBINY/(	106)	STORE	M	TB55
TB56		M	Stored booster value of wgross	/ORBINY/(	107)	STORE WTVOL	M M	TB56 TB56
TB6		M	Stored booster value of clbody	/ORBINY/(	11)	STORE	M	TB6
TOVERC		M	Wing thickness over choord ratio	/CINPUT/(	388)	PROTHR STORE WTSCH	I M I	TOVERC TOVERC TOVERC
TO10		M	Stored orbiter value of csfair	/ORBINX/(	15)	STORE	M	TO10
TO11		M	Stored orbiter value of csfutk	/ORBINX/(	16)	STORE	M	TO11
TO12		M	Stored orbiter value of csoxtk	/ORBINX/(	17)	STORE	M	TO12
TO13		M	Stored orbiter value of cshorz	/ORBINX/(	18)	STORE	M	TO13
TO15		M	Stored orbiter value of csplan	/ORBINX/(	20)	STORE	M	TO15
TO16		M	Stored orbiter value of csvert	/ORBINX/(	21)	STORE	M	TO16
TO17		M	Stored orbiter value of cswing	/ORBINX/(	22)	STORE	M	TO17
TO18		M	Stored orbiter value of cthrst	/ORBINX/(	23)	STORE WTVOL	M M	TO18 TO18
TO19		M	Stored orbiter value of cthstz	/ORBINX/(	24)	STORE	M	TO19
TO2		M	Stored orbiter value of cbbody	/ORBINX/(	2)	STORE	M	TO2
TO27		M	Stored orbiter value of isp(i)	/ORBINX/(	41)	SIZEMR SSSP STORE SUMOUT VEHOF WTVOL	I I M I M I	TO27 TO27 TO27 TO27 TO27 TO27
TO34		M	Stored orbiter value of mr(i)	/ORBINX/(	53)	ITERB SSSP STORE TAMPER VEHOF WTVOL	M O M I O M	TO34 TO34 TO34 TO34 TO34 TO34
TO35		M	Stored orbiter value of ncrew	/ORBINX/(	59)	STORE	M	TO35
TO36		M	Stored orbiter value of nengs	/ORBINX/(	60)	STORE SUMOUT TAMPER THRUST WTVOL	M I I I I	TO36 TO36 TO36 TO36 TO36
TO38		M	Stored orbiter value of npass	/ORBINX/(	62)	STORE	M	TO38
TO4		M	Stored orbiter value of cfuel(i)	/ORBINX/(	4)	STORE	M	TO4
TO41		M	Stored orbiter value of rhofu	/ORBINX/(	65)	STORE	M	TO41
TO42		M	Stored orbiter value of rhofu2	/ORBINX/(	66)	STORE	M	TO42
TO43		M	Stored orbiter value of rhox	/ORBINX/(	67)	STORE	M	TO43
TO45		M	Stored orbiter value of swing	/ORBINX/(	69)	STORE SUMOUT TAMPER	M I I	TO45 TO45 TO45
TO47		M	Stored orbiter value of toverc	/ORBINX/(	71)	STORE	M	TO47
TO5		M	Stored orbiter value of chbody	/ORBINX/(	10)	STORE	M	TO5

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBR	CODE	VAR
T051		M	Stored orbiter value of vbody	/ORBINX/(	102)	STORE WTVOL	M O	T051 T051
T052		O	Stored orbiter value of vfutk	/ORBINX/(	103)	STORE	O	T052
T053		M	Stored orbiter value of vfutk2	/ORBINX/(	104)	STORE	M	T053
T054		O	Stored orbiter value of voxtk	/ORBINX/(	105)	STORE	O	T054
T055		M	Stored orbiter value of voxtk2	/ORBINX/(	106)	STORE	M	T055
T056		M	Stored orbiter value of mgross	/ORBINX/(	107)	STORE WTVOL	M M	T056 T056
T06		M	Stored orbiter value of clbody	/ORBINX/(	11)	STORE	M	T06
T066		M	Working name for input k-array orbiter volume scaling coeff	/ORBINX/(	114)	STORE	M	T066
T07		M	Stored orbiter value of csbody	/ORBINX/(	12)	STORE	M	T07
TPRATO		M	Wing taper ratio	/CINPUT/(	389)	STORE WTSCH	M I	TPRATO TPRATO
TTOT		I	Total stage vac. Thrust	/VOLCAL/(	21)	PRITVA STORE WTSCH WTVOL	I I M I	TTOT TTOT TTOT TTOT
VBODY		M	Total body volume	/CINPUT/(	391)	PRINTV SOLVE STORE TAMPER WTSCH WTVOL	M M M I M I	VBODY VBODY VBODY VBODY VBODY VBODY
VBODYR		W	Vbody to - 2/3 power	/STORE /(*	)	STORE	W	VBODYR
VFUTK		M	Total volume of fuel tank	/VOLCAL/(	29)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK VFUTK VFUTK VFUTK
VFUTK2		M	Total volume of secondary fuel tank	/VOLCAL/(	30)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK2 VFUTK2 VFUTK2 VFUTK2
VOXTK		M	Total volume of oxidizer tank	/VOLCAL/(	34)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK VOXTK VOXTK VOXTK
VOXTK2		M	Total volume of secondary oxidizer tank	/VOLCAL/(	35)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK2 VOXTK2 VOXTK2 VOXTK2
WACRES		M	Weight of attitude control fuel reserve	/WTCALC/(	5)	PRINTW STORE WTSCH	I M M	WACRES WACRES WACRES
WACSFO		M	Weight of attitude control fuel plus oxidizer	/WTCALC/(	7)	PRINTW STORE WTSCH	I M M	WACSFO WACSFO WACSFO
WAUXT		I	Weight of separation system	/WTCALC/(	10)	PRINTW STORE WTSCH	I I M	WAUXT WAUXT WAUXT
WDECAY		M	Thrust decay propellant weight	/WTCALC/(	18)	PRINTW STORE WTSCH	I M M	WDECAY WDECAY WDECAY
WFROST		M	Frost and ice weight	/WTCALC/(	33)	PRINTW STORE WTSCH	I M M	WFROST WFROST WFROST
WFULOS		M	Vented fuel	/WTCALC/(	44)	PRINTW STORE WTSCH	I M M	WFULOS WFULOS WFULOS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WFURES		M	Fuel reserve	/WTCALC/(	47)	PRINTW	I	WFURES
						STORE	M	WFURES
						TAMPER	I	WFURES
						WTSCH	M	WFURES
WFUTRP		M	Trapped fuel weight	/WTCALC/(	52)	PRINTW	I	WFUTRP
						STORE	M	WFUTRP
						WTSCH	M	WFUTRP
WGASPR		M	Weight of gas and pressurant	/WTCALC/(	53)	PRINTW	I	WGASPR
						STORE	M	WGASPR
						WTSCH	M	WGASPR
WGROSS		M	Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
						PRWTSM	I	WGROSS
						SOLVE	M	WGROSS
						STORE	M	WGROSS
						TAMPER	I	WGROSS
						WTSCH	M	WGROSS
						WTVOL	I	WGROSS
WJET		O	Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA	I	WJET
						PRWTSM	M	WJET
						STORE	O	WJET
						TAMPER	I	WJET
						WTSCH	M	WJET
WOXLOS		M	Vented oxidizer	/WTCALC/(	88)	PRINTW	I	WOXLOS
						STORE	M	WOXLOS
						WTSCH	M	WOXLOS
WOXRES		M	Oxidizer reserve	/WTCALC/(	89)	PRINTW	I	WOXRES
						STORE	M	WOXRES
						TAMPER	I	WOXRES
						WTSCH	M	WOXRES
WOXTRP		M	Trapped oxidizer weight	/WTCALC/(	94)	PRINTW	I	WOXTRP
						STORE	M	WOXTRP
						WTSCH	M	WOXTRP
WPOWFO		M	Power system propellant wt.	/WTCALC/(	101)	PRINTW	I	WPOWFO
						STORE	M	WPOWFO
						WTSCH	M	WPOWFO
WPOWRS		M	Power system propellant reserve	/WTCALC/(	102)	PRINTW	I	WPOWRS
						STORE	M	WPOWRS
						WTSCH	M	WPOWRS
WSRTRP		M	Trapped oxidizer weight	/WTCALC/(	114)	PRINTW	I	WSRTRP
						STORE	M	WSRTRP
						WTSCH	M	WSRTRP
WWAIT		M	Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA	I	WWAIT
						PRWTSM	I	WWAIT
						STORE	M	WWAIT
						TAMPER	I	WWAIT
						WTSCH	M	WWAIT

```
      SUBROUTINE STORE                                                  STORE
C                                                                       STORE
C     SYNTHESIS CONTROL SUBROUTINE FOR TWO-STAGE VEHICLES               STORE
C                                                                       STORE
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      COMMON/WTCALC/   ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                    CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,    ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27, ORBINX
     3 TO28,TO29,TO30,TO31,TO32,TO33,TO34,TO35,TO36,TO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                         CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,     UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,     UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                    CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,    ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27, ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                               UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,     UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,      UH
     8TB83,TB84                                                         UH
C                                                                       STORE
      SWING=CSWING                                                      STORE
      IF(CSHORZ.GT.20.) CSHORZ=CSHORZ/SWING                             STORE
      VBODYR = 1./(VBODY**.666667)                                      STORE
      IF(CSBODY.GT.20.) CSBODY=CSBODY*VBODYR                            STORE
      IF(CSPLAN.GT.20.) CSPLAN=CSPLAN*VBODYR                            STORE
      IF(CSVERT.GT.5.) CSVERT=CSVERT*VBODYR                             STORE
      IF(CSFAIR.GT.20.) CSFAIR=CSFAIR/(CSBODY/VBODYR)                   STORE
      VBODYR = SQRT(VBODYR)                                             STORE
      IF(CBBODY.GT.5.) CBBODY = CBBODY*VBODYR                           STORE
      IF(CHBODY.GT.5.) CHBODY = CHBODY*VBODYR                           STORE
      IF(CLBODY.GT.20.) CLBODY=CLBODY *VBODYR                           STORE
      RETURN                                                            STORE
C***************************************************************        STORE
```

```
C      ORBITER DATA  STORAGE                                            STORE
C*********************************************************************** STORE
       ENTRY ORBSTO                                                     STORE
       T02 = CBBODY                                                     STORE
       DO 20 I = 1,6                                                    STORE
    20 T04(I) = CFUEL(I)                                                STORE
       T05 = CHBODY                                                     STORE
       T06 = CLBODY                                                     STORE
       T07 = CSBODY                                                     STORE
       T010 = CSFAIR                                                    STORE
       T011 = CSFUTK                                                    STORE
       T012 = CSOXTK                                                    STORE
       T013 = CSHORZ                                                    STORE
       T015 = CSPLAN                                                    STORE
       T016 = CSVERT                                                    STORE
       T017 = CSWING                                                    STORE
       T018 = CTHRST                                                    STORE
       T019 = CTHST2                                                    STORE
       DO 24 I = 1,6                                                    STORE
    24 T027(I) = ISP(I)                                                 STORE
       DO 26 I = 1,6                                                    STORE
    26 T034(I) = MR(I)                                                  STORE
       T035 = NCREW                                                     STORE
       T036 = NENGS                                                     STORE
       T038 = NPASS                                                     STORE
       T041 = RHOFU                                                     STORE
       T042 = RHOFU2                                                    STORE
       T043 = RHOX                                                      STORE
       T044 = RHOX2                                                     STORE
       T045 = SWING                                                     STORE
       T047 = TOVERC                                                    STORE
       T051 = VBODY                                                     STORE
       T052 = VFUTK                                                     STORE
       T053 = VFUTK2                                                    STORE
       T054 = VOXTK                                                     STORE
       T055 = VOXTK2                                                    STORE
       T056 = WGROSS                                                    STORE
       DO 30 I = 1,30                                                   STORE
    30 SKO(I)= K(I)                                                     STORE
       DO 50 I = 1,300                                                  STORE
    50 SCO(I)= C(I)                                                     STORE
       T059 = LF                                                        STORE
       T060=TPRATO                                                      STORE
       T061=ASRATO                                                      STORE
       T062=FXWOVS                                                      STORE
       T063=NWL                                                         STORE
       T064=ITPS                                                        STORE
       T065=PCHAM                                                       STORE
       T066=ANENGS                                                      STORE
       T067=ANTANK                                                      STORE
       T068=TYTAIL                                                      STORE
       T069=NLISTO                                                      STORE
       T070=ASWEEP                                                      STORE
       T071=WFROST                                                      STORE
       T072=WFUTRP                                                      STORE
       T073=WOXTRP                                                      STORE
       T074=WSRTRP                                                      STORE
       T075=WDECAY                                                      STORE
       T076=WFURES                                                      STORE
       T077=WOXRES                                                      STORE
       T078=WACSFO                                                      STORE
       T079=WFULOS                                                      STORE
       T080=WOXLOS                                                      STORE
       T081=WPOWFO                                                      STORE
       T082=WGASPR                                                      STORE
       T083=WACRES                                                      STORE
       T084=WPOWRS                                                      STORE
       DO 999 I=1,10                                                    STORE
   999 OWSAVE(I)=WWAIT(I)                                               STORE
       RETURN                                                           STORE
C*********************************************************************** STORE
C      BOOSTER DATA STORAGE                                             STORE
C*********************************************************************** STORE
       ENTRY BOOSTO                                                     STORE
       TB2 = CBBODY                                                     STORE
```

1541

```
      DO 60 I = 1,6                                                     STORE
   60 TB4(I) = CFUEL(I)                                                 STORE
      TB5 = CHBODY                                                      STORE
      TB6 = CLBODY                                                      STORE
      TB7 = CSBODY                                                      STORE
      TB10 = CSFAIR                                                     STORE
      TB11 = CSFUTK                                                     STORE
      TB12 = CSOXTK                                                     STORE
      TB13 = CSHORZ                                                     STORE
      TB15 = CSPLAN                                                     STORE
      TB16 = CSVERT                                                     STORE
      TB17 = CSWING                                                     STORE
      TB18 = CTHRST                                                     STORE
      TB19 = CTHST2                                                     STORE
      DO 64 I = 1,6                                                     STORE
   64 TB27(I) = ISP(I)                                                  STORE
      DO 66 I = 1,6                                                     STORE
   66 TB34(I) = MR(I)                                                   STORE
      TB35 = NCREW                                                      STORE
      TB36 = NENGS                                                      STORE
      TB38 = NPASS                                                      STORE
      TB41 = RHOFU                                                      STORE
      TB42 = RHOFU2                                                     STORE
      TB43 = RHOX                                                       STORE
      TB44 = RHOX2                                                      STORE
      TB45 = SWING                                                      STORE
      TB47 = TOVERC                                                     STORE
      TB51 = VBODY                                                      STORE
      TB52 = VFUTK                                                      STORE
      TB53 = VFUTK2                                                     STORE
      TB54 = VOXTK                                                      STORE
      TB55 = VOXTK2                                                     STORE
      TB56 = WGROSS                                                     STORE
      DO 80 I = 1, 30                                                   STORE
   80 SKB(I) = K(I)                                                     STORE
      DO 100 I = 1, 300                                                 STORE
  100 SCB(I) = C(I)                                                     STORE
      TB59 = LF                                                         STORE
      TB60=TPRATO                                                       STORE
      TB61=ASRATO                                                       STORE
      TB62=FXWOVS                                                       STORE
      TB63=NWL                                                          STORE
      TB64=ITPS                                                         STORE
      TB65=PCHAM                                                        STORE
      TB66=ANENGS                                                       STORE
      TB67=ANTANK                                                       STORE
      TB68=TYTAIL                                                       STORE
      TB69=NLISTO                                                       STORE
      TB70=ASWEEP                                                       STORE
      TB71=WFROST                                                       STORE
      TB72=WFUTRP                                                       STORE
      TB73=WOXTRP                                                       STORE
      TB74=WSRTRP                                                       STORE
      TB75=WDECAY                                                       STORE
      TB76=WFURES                                                       STORE
      TB77=WOXRES                                                       STORE
      TB78=WACSFO                                                       STORE
      TB79=WFULOS                                                       STORE
      TB80=WOXLOS                                                       STORE
      TB81=WPOWFO                                                       STORE
      TB82=WGASPR                                                       STORE
      TB83=WACRES                                                       STORE
      TB84=WPOWRS                                                       STORE
      DO 997 I=1,10                                                     STORE
  997 BWSAVE(I)=WWAIT(I)                                                STORE
      RETURN                                                            STORE
C*********************************************************************** STORE
C     ORBITER DATA RETREIVAL                                            STORE
C*********************************************************************** STORE
      ENTRY ORBCAL                                                      STORE
      CBBODY    = T02                                                   STORE
      DO 200I = 1,6                                                     STORE
  200 CFUEL(I) = T04(I)                                                 STORE
      CHBODY    = T05                                                   STORE
      CLBODY    = T06                                                   STORE
```

1542

```
      CSBODY    = T07                                              STORE
      CSFAIR    = T010                                             STORE
      CSFUTK    = T011                                             STORE
      CSOXTK    = T012                                             STORE
      CSHORZ    = T013                                             STORE
      CSPLAN    = T015                                             STORE
      CSVERT    = T016                                             STORE
      CSWING    = T017                                             STORE
      CTHRST    = T018                                             STORE
      CTHST2    = T019                                             STORE
      DO 204I = 1,6                                                STORE
  204 ISP(I)    = T027(I)                                          STORE
      DO 206I = 1,6                                                STORE
  206 MR(I)     = T034(I)                                          STORE
      NCREW     = T035                                             STORE
      NENGS     = T036                                             STORE
      NPASS     = T038                                             STORE
      RHOFU     = T041                                             STORE
      RHOFU2    = T042                                             STORE
      RHOX      = T043                                             STORE
      RHOX2     = T044                                             STORE
      SWING     = T045                                             STORE
      TOVERC    = T047                                             STORE
      VBODY     = T051                                             STORE
      VFUTK     = 0.                                               STORE
      VFUTK2    = T053                                             STORE
      VOXTK     = 0.                                               STORE
      VOXTK2    = T055                                             STORE
      WGROSS    = T056                                             STORE
      DO 230 I = 1,30                                              STORE
  230 K(I) = SKD(I)                                                STORE
      DO 260 I = 1,300                                             STORE
  260 C(I) = SCO(I)                                                STORE
      LF = T059                                                    STORE
      TPRATO=T060                                                  STORE
      ASRATO=T061                                                  STORE
      FXWOVS=T062                                                  STORE
      NWL=T063                                                     STORE
      ITPS=T064                                                    STORE
      PCHAM=T065                                                   STORE
      ANENGS=T066                                                  STORE
      ANTANK =T067                                                 STORE
      TYTAIL=T068                                                  STORE
      NLISTO=T069                                                  STORE
      ASWEEP=T070                                                  STORE
      WFROST=T071                                                  STORE
      WFUTRP=T072                                                  STORE
      WOXTRP=T073                                                  STORE
      WSRTRP=T074                                                  STORE
      WDECAY=T075                                                  STORE
      WFURES=T076                                                  STORE
      WOXRES=T077                                                  STORE
      WACSFO=T078                                                  STORE
      WFULOS=T079                                                  STORE
      WOXLOS=T080                                                  STORE
      WPOWFO=T081                                                  STORE
      WGASPR=T082                                                  STORE
      WACRES=T083                                                  STORE
      WPOWRS=T084                                                  STORE
      DO 996 I=1,10                                                STORE
  996 WWAIT(I)=OWSAVE(I)                                           STORE
      RETURN                                                       STORE
      ENTRY BOOCAL                                                 STORE
C************************************************************************ STORE
C     BOOSTER DATA RETREIVAL                                       STORE
C************************************************************************ STORE
      CBBODY    = TB2                                              STORE
      DO 300I = 1,6                                                STORE
  300 CFUEL(I) = TB4(I)                                            STORE
      CHBODY    = TB5                                              STORE
      CLBODY    = TB6                                              STORE
      CSBODY    = TB7                                              STORE
      CSFAIR    = TB10                                             STORE
      CSFUTK    = TB11                                             STORE
      CSOXTK    = TB12                                             STORE
```

20 OCT 72 6.01-46

```
      CSHORZ   = TB13                                                   STORE
      CSPLAN   = TB15                                                   STORE
      CSVERT   = TB16                                                   STORE
      CSWING   = TB17                                                   STORE
      CTHRST   = TB18                                                   STORE
      CTHST2   = TB19                                                   STORE
      DO 304I = 1,6                                                     STORE
  304 ISP(I)   = TB27(I)                                                STORE
      DO 306I = 1,6                                                     STORE
  306 MR(I)    = TB34(I)                                                STORE
      NCREW    = TB35                                                   STORE
      NENGS    = TB36                                                   STORE
      NPASS    = TB38                                                   STORE
      RHOFU    = TB41                                                   STORE
      RHOFU2   = TB42                                                   STORE
      RHOX     = TB43                                                   STORE
      RHOX2    = TB44                                                   STORE
      SWING    = TB45                                                   STORE
      TOVERC   = TB47                                                   STORE
      VBODY    = TB51                                                   STORE
      VFUTK    = 0.                                                     STORE
      VFUTK2   = TB53                                                   STORE
      VOXTK    = 0.                                                     STORE
      VOXTK2   = TB55                                                   STORE
      DO 400 I = 1,30                                                   STORE
  400 K(I) = SKB(I)                                                     STORE
      DO 450 I = 1,300                                                  STORE
  450 C(I) = SCB(I)                                                     STORE
      LF = TB59                                                         STORE
      TPRATO=TB60                                                       STORE
      ASRATO=TB61                                                       STORE
      FXWOVS=TB62                                                       STORE
      NWL=TB63                                                          STORE
      ITPS=TB64                                                         STORE
      PCHAM=TB65                                                        STORE
      ANENGS=TB66                                                       STORE
      ANTANK =TB67                                                      STORE
      TYTAIL=TB68                                                       STORE
      NLISTO=TB69                                                       STORE
      ASWEEP=TB70                                                       STORE
      WFROST=TB71                                                       STORE
      WFUTRP=TB72                                                       STORE
      WOXTRP=TB73                                                       STORE
      WSRTRP=TB74                                                       STORE
      WDECAY=TB75                                                       STORE
      WFURES=TB76                                                       STORE
      WOXRES=TB77                                                       STORE
      WACSFO=TB78                                                       STORE
      WFULOS=TB79                                                       STORE
      WOXLOS=TB80                                                       STORE
      WPOWFO=TB81                                                       STORE
      WGASPR=TB82                                                       STORE
      WACRES=TB83                                                       STORE
      WPOWRS=TB84                                                       STORE
      DO 995 I=1,10                                                     STORE
  995 WWAIT(I)=BWSAVE(I)                                                STORE
      IF(TB18.EQ 0 .AND.SCB(129).EQ.0.) C(129) = TTOT * NENGS           STORE
      IF(SCB(153).EQ.0.) C(154) = WAUXT * SCB(263)                      STORE
      C(105)=WGROSS                                                     STORE
      WGROSS   = TB56                                                   STORE
      WJET(3)=C(105)+C(78)                                              STORE
      RETURN                                                            STORE
      END                                                               STORE
```

SUBRØUTINE SUMØUT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
BTTOT		I	Total booster weight flow	/TAMP /(	5)	SUMOUT	I	BTTOT
						TAMPER	I	BTTOT
						WTVOL	O	BTTOT
FRANK		W	Total ascent propellant weight including fpr	/SUMOUT/(*	)	SUMOUT	W	FRANK
IFIRE		W	Integer value of fire flag	/SUMOUT/(*	)	SUMOUT	W	IFIRE
OTTOT		I	Total orbiter thrust	/TAMP /(	2)	SUMOUT	I	OTTOT
						TAMPER	I	OTTOT
						WTVOL	M	OTTOT
SBODYB		I	Total body wetted area- booster	/SUMVW /(	13)	SUMOUT	I	SBODYB
						TAMPER	O	SBODYB
SBODYO		I	Total body watteo area- orbiter	/SUMVW /(	26)	SUMOUT	I	SBODYO
						TAMPER	O	SBODYO
SCB		I	Working name for input c-array booster scaling coefficients	/ORBINY/(	144)	FLYBKP	M	SCB
						STORE	M	SCB
						SUMOUT	I	SCB
						TAMPER	I	SCB
						THRUST	M	SCB
						VEHDF	I	SCB
						WTVOL	M	SCB
SE		I	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SPLANB		I	Booster body planform area	/SUMVW /(	14)	SUMOUT	I	SPLANB
						TAMPER	O	SPLANB
SPLANO		I	Orbiter body planform area	/SUMVW /(	27)	SUMOUT	I	SPLANO
						TAMPER	O	SPLANO
SPSLN		W	Nominal specific impulse- sea level	/SUMOUT/(*	)	SUMOUT	W	SPSLN
SPVAN		W	Nominal specific impulse- vacuum	/SUMOUT/(*	)	SUMOUT	W	SPVAN
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SUMOUT		E	Subroutine to print summary data and calculate thrust for output purposes only	/SUMOUT/(S	)	SSSP	S	SUMOUT
						SUMOUT	E	SUMOUT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB27		I	Stored booster value of isp(I)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		I	Stored booster value of mr(I)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TB36		I	Stored booster value of mengs	/ORBINY/(	60)	STORE	M	TB36
						SUMOUT	I	TB36
						TAMPER	I	TB36
						THRUST	I	TB36
						WTVOL	I	TB36
TB45		I	Stored booster value of swing	/ORBINY/(	69)	STORE	M	TB45
						SUMOUT	I	TB45
						TAMPER	I	TB45
TO27		I	Stored orbiter value of isp(I)	/ORBINX/(	41)	SIZEMR	I	TO27
						SSSP	I	TO27
						STORE	M	TO27
						SUMOUT	I	TO27
						VEHDF	M	TO27
						WTVOL	I	TO27
TO36		I	Stored orbiter value of mengs	/ORBINX/(	60)	STORE	M	TO36
						SUMOUT	I	TO36
						TAMPER	I	TO36
						THRUST	I	TO36
						WTVOL	I	TO36
TO45		I	Stored orbiter value of swing	/ORBINX/(	69)	STORE	M	TO45
						SUMOUT	I	TO45
						TAMPER	I	TO45
TSLN		W	Nominal sea level thrust for output	/SUMOUT/(*	)	SUMOUT	W	TSLN
TSLNB		W	Nominal sea level thrust for output booster	/SUMOUT/(*	)	SUMOUT	W	TSLNB
TSLNO		W	Nominal sea level thrust for output orbiter	/SUMOUT/(*	)	SUMOUT	W	TSLNO
TVAN		W	Nominal vac. Thrust-output (vehicle)	/SUMOUT/(*	)	SUMOUT	W	TVAN
TVANB		W	Nominal vac. Thrust-output (booster)	/SUMOUT/(*	)	SUMOUT	W	TVANB
TVANO		W	Nominal vac. Thrust-output (orbiter)	/SUMOUT/(*	)	SUMOUT	W	TVANO
TWORB		W	Orbiter thrust-to-weight ratio	/SUMOUT/(*	)	SUMOUT	W	TWORB
VBODYB		I	Booster body volume	/SUMVW /(	10)	SUMOUT	I	VBODYB
						TAMPER	O	VBODYB
VBODYO		I	Orbiter body volume	/SUMVW /(	24)	SUMOUT	I	VBODYO
						TAMPER	O	VBODYO
VCARO		I	Volume of cargo	/SUMVW /(	22)	SUMOUT	I	VCARO
						TAMPER	O	VCARO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
VFUTKB		I	Total volume of fuel tank - booster	/SUMVW /(	7)	SUMOUT TAMPER	I O	VFUTKB VFUTKB
VFUTKO		I	Total volume of fuel tank - orbiter	/SUMVW /(	20)	SUMOUT TAMPER	I O	VFUTKO VFUTKO
VOTHB		I	Misc. Booster volume	/SUMVW /(	9)	SUMOUT TAMPER	I O	VOTHB VOTHB
VOTHO		I	Misc. Orbiter volume	/SUMVW /(	23)	SUMOUT TAMPER	I O	VOTHO VOTHO
VOXTKB		I	Booster oxidizer tank volume	/SUMVW /(	8)	SUMOUT TAMPER	I O	VOXTKB VOXTKB
VOXTKO		I	Orbiter oxidizer tank volume	/SUMVW /(	21)	SUMOUT TAMPER	I O	VOXTKO VOXTKO
VPROPB		W	Volume of propulsion bay - booster	/SUMOUT/(*	)	SUMOUT	W	VPROPB
VPROPO		W	Volume of propulsion bay - orbiter	/SUMOUT/(*	)	SUMOUT	W	VPROPO
WABFUB		I	Booster flyback fuel required	/SUMVW /(	2)	SUMOUT TAMPER	I M	WABFUB WABFUB
WABFUO		I	Orbiter flyback fuel required	/SUMVW /(	19)	SUMOUT TAMPER	I M	WABFUO WABFUO
WBPTX		W	Ascent propellant weight - booster	/SUMOUT/(*	)	SUMOUT	W	WBPTX
WCONTB		I	Contingency and growth weight-boostr	/SUMVW /(	38)	SUMOUT TAMPER	I O	WCONTB WCONTB
WCONTO		I	Contingency and growth weight-orbitr	/SUMVW /(	37)	SUMOUT TAMPER	I O	WCONTO WCONTO
WDOTN		W	Nominal weight flow	/SUMOUT/(*	)	SUMOUT	W	WDOTN
WDRYB		I	Stage dry weight - booster	/SUMVW /(	4)	SUMOUT TAMPER	I O	WDRYB WDRYB
WDRYO		I	Stage dry weight -orbiter	/SUMVW /(	17)	SUMOUT TAMPER	I O	WDRYO WDRYO
WENTRB		I	Booster entry weight	/SUMVW /(	34)	SUMOUT TAMPER	I O	WENTRB WENTRB
WENTRO		I	Orbiter entry weight	/SUMVW /(	33)	SUMOUT TAMPER	I O	WENTRO WENTRO
WFUOXB		I	Propellant wt. Less fpr -booster	/SUMVW /(	3)	SUMOUT TAMPER	I M	WFUOXB WFUOXB
WFUOXO		I	Propellant wt. Less fpr -orbiter	/TAMP /(	3)	SUMOUT TAMPER WTVOL	I I M	WFUOXO WFUOXO WFUOXO
WGROSB		I	Booster gross weight	/SUMVW /(	5)	SUMOUT TAMPER	I M	WGROSB WGROSB
WGROSO		I	Orbiter gross weight	/TAMP /(	1)	SSSP SUMOUT WTVOL	I I M	WGROSO WGROSO WGROSO
WLANDB		I	Landing weight - booster	/SUMVW /(	35)	SUMOUT TAMPER	I O	WLANDB WLANDB
WLANDO		I	Landing weight - orbiter	/SUMVW /(	36)	SUMOUT TAMPER	I O	WLANDO WLANDO
WOP		I	Orbit maneuvering propellant weight	/SUMVW /(	1)	SUMOUT TAMPER	I O	WOP WOP
WOPT		W	Total ascent propellant weight	/SUMOUT/(*	)	SUMOUT	W	WOPT
WOPTX		W	Ascent propellant for orbiter including fpr	/SUMOUT/(*	)	SUMOUT	W	WOPTX
WORBTO		I	In-orbit weight - orbiter	/SUMVW /(	29)	SUMOUT TAMPER	I O	WORBTO WORBTO
WOTHB		I	Misc Weight - booster	/SUMVW /(	6)	SUMOUT TAMPER	I O	WOTHB WOTHB
WOTHO		I	Misc. Weight - orbiter	/SUMVW /(	18)	SUMOUT TAMPER	I O	WOTHO WOTHO
WOVRSB		I	Booster wing loading	/SUMVW /(	15)	SUMOUT TAMPER	I O	WOVRSB WOVRSB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WOVRSO		I	Orbiter wing loading	/SUMVW /(	28)	SUMOUT	I	WOVRSO
						TAMPER	O	WOVRSO
WPAYLO		I	Payload weight	/SUMVW /(	16)	SSSP	I	WPAYLO
						SUMOUT	I	WPAYLO
						TAMPER	M	WPAYLO
WRTRNB		I	Entry weight- booster	/SUMVW /(	32)	SUMOUT	I	WRTRNB
						TAMPER	O	WRTRNB
WRTRNO		I	Entry weight- orbiter	/SUMVW /(	31)	SUMOUT	I	WRTRNO
						TAMPER	O	WRTRNO
.UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITERB	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEQ	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE SUMOUT                                                 SUMOUT
C                                                                       SUMOUT
C         SUBROUTINE TO PRINT SUMMARY DATA                              SUMOUT
C         CALCULATES THRUST FOR OUTPUT PURPOSES ONLY                    SUMOUT
C                                                                       SUMOUT
      REAL LBODYO,LBODYB                                                SUMOUT
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),     QP(14),    EROR,     PZ(5),      VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,      TLNG,                   SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,      DWEB,      DWEO,     TOLWT,    WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,    TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,    IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IOVEL,    ISPO,    ISPB,   SIZING
     *XPL,     TVACB,    NNB,        WEO,      WEB,      WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,       MUO,      VSTG,     WPO             SIZING
     *,JTYP,    BECO ,    BSTG ,     ORBI ,   ITNBW , ITNOW  ,            SIZING
     * SVDPSQ  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZD(19)                      UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                    CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,    ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,IO22,IO23,IO24,IO25,IO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                          CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,     UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,     UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                    CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,    ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                               UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,     UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,      UH
     8TB83,TB84                                                         UH
      COMMON/TAMP/WGROSO,OTTOT,WFUOXO,TBTO,BTTOT                        SUMOUT
      COMMON/SUMVW/WOP,            WABFUB,WFUOXB,WDRYB,WGROSB,WOTHB,    SUMOUT
     1 VFUTKB,VOXTKB,VOTHB,VBODYB,VABFUB,                               SUMOUT
     2 LBODYB,SBODYB,SPLANB,WOVRSB,                                     SUMOUT
     1                  WPAYLO,WDRYO,WOTHO,  WABFUO,                    SUMOUT
     2 VFUTKO,VOXTKO,VCARO,VOTHO,VBODYO,                                SUMOUT
     3 LBODYO,SBODYO,SPLANO,WOVRSO                                      SUMOUT
     4 ,WORBTO,WORBTB,WRTRNO,WRTRNB,WENTRO,WENTRB,WLANOB,WLANOO         SUMOUT
     5 ,  WCONTO,WCONTB                                                 SUMOUT
      IFIRE = SE(2)                                                     SUMOUT
      SQ(14,3) = SQ(14,2) - SQ(14,1)                                    SUMOUT
      FRANK = WFUOXO - WOP                                              SUMOUT
      TWORB = OTTOT/WGROSO                                              SUMOUT
      WORYT =     WDRYB + WDRYO                                         SUMOUT
      WGROSV =     WGROSB + WGROSO                                      SUMOUT
      VPROPO = VFUTKO + VOXTKO                                          SUMOUT
      VPROPB = VFUTKB + VOXTKB                                          SUMOUT
   30 TVANO = OTTOT/TO36                                                SUMOUT
      TVANB =  BTTOT/TB36                                               SUMOUT
      TSLNO = TVANO * TO27(2)/TO27(3)                                   SUMOUT
      TSLNB = TVANB * TB27(2)/TB27(3)                                   SUMOUT
      X = 1.                                                            SUMOUT
      IF(SE(8).EQ.0..OR.IFIRE.EQ.2)  X=0.                               SUMOUT
      TVAN = OTTOT * X +  BTTOT                                         SUMOUT
      WDOTN = OTTOT * X /TO27(3) +  BTTOT / TB27(3)                     SUMOUT
      SPVAN = TVAN/WDOTN                                                SUMOUT
      TSLN = TSLNO * TO36 * X + TSLNB * TB36                            SUMOUT
      SPSLN = TSLN/WDOTN                                                SUMOUT
      WOPT = FRANK + SQ(37,4)                                           SUMOUT
      WOPTX = WFUOXO + SQ(37,4)                                         SUMOUT
      WBPTX = WFUOXB + SQ(37,5)                                         SUMOUT
      NCOPIE = SV(15)+ .1                                               SUMOUT
      DO 900 J=1,NCOPIE                                                 SUMOUT
```

```
      WRITE (6,1000)                                                    SUMOUT
      WRITE (6,1005)                                                    SUMOUT
      WRITE (6,1010)                                                    SUMOUT
      WRITE(6,1011) WFUOXB, FRANK                                       SUMOUT
      WRITE (6,1012) SQ(37,5), SQ(37,4)                                 SUMOUT
      WRITE (6,1015) WBPTX,WGPT                                         SUMOUT
      WRITE (6,1020)  WOP                                               SUMOUT
      WRITE (6,1025) WBPTX,WOPTX                                        SUMOUT
      WRITE (6,1022) WABFUB ,WABFUO                                     SUMOUT
      WRITE (6,1030)  WPAYLO                                            SUMOUT
      WRITE (6,1035)  WDRYB,WDRYO ,WDRYT                                SUMOUT
      WRITE (6,1041) WCONTB,WCONTO                                      SUMOUT
      WRITE (6,1040)  WOTHB,WOTHO                                       SUMOUT
      WRITE (6,1045)  WGROSB,WGROSO,WGROSV                              SUMOUT
      WRITE (6,1170)           WORBTO                                   SUMOUT
      WRITE (6,1175) WRTANB,WRTANO                                      SUMOUT
      WRITE (6,1180) WENTRB,WENTRO                                      SUMOUT
      WRITE (6,1185) WLANDB,WLANDO                                      SUMOUT
      WRITE (6,1050)                                                    SUMOUT
      WRITE (6,1055)  VFUTKB,VFUTKO                                     SUMOUT
      WRITE (6,1060)  VOXTKB,VOXTKO                                     SUMOUT
      WRITE (6,1065)  VPROPB,VPROPO                                     SUMOUT
      WRITE (6,1070)  VCARO                                             SUMOUT
      WRITE (6,1075)  VOTHB,VOTHO                                       SUMOUT
      WRITE (6,1077) VBODYB,VBODYO                                      SUMOUT
      WRITE (6,1080)                                                    SUMOUT
      WRITE (6,1085)  LBODYB,LBODYO                                     SUMOUT
      WRITE (6,1090)  SBODYB,SBODYO                                     SUMOUT
      WRITE (6,1095)  SPLANB,SPLANO                                     SUMOUT
      WRITE (6,1096) TB45, TO45                                         SUMOUT
      WRITE (6,1100)  WOVRSB,WOVRSO                                     SUMOUT
      WRITE (6,1105)                                                    SUMOUT
      WRITE (6,1110) TWORB,SV(13)                                       SUMOUT
      WRITE (6,1111) TB36,TO36                                          SUMOUT
      IF(SE(8).EQ.0..OR. IFIRE.EQ.2) GO TO 200                          SUMOUT 200
      WRITE (6,1112) TSLNB, TSLNO, TSLN                                 SUMOUT
      WRITE (6,1113) TVANB, TVANO, TVAN                                 SUMOUT
      WRITE (6,1115) SPSLN                                              SUMOUT
      WRITE (6,1120) SPVAN                                              SUMOUT
C     NO CROSSFEED                                                      SUMOUT
  200 CONTINUE                                                          SUMOUT
      WRITE (6,2112) TSLNB, TSLNO, TSLN                                 SUMOUT
      WRITE (6,2113) TVANB, TVANO, TVAN                                 SUMOUT
      WRITE (6,2115) SPSLN                                              SUMOUT
      WRITE (6,2120) SPVAN                                              SUMOUT
  220 WRITE (6,1125)                                                    SUMOUT
      WRITE (6,1126)      TB34(3),SV(6)                                 SUMOUT
      WRITE(6,1128) SQ(14,1),SQ(14,3),SQ(14,2)                          SUMOUT
      WRITE (6,1127) SE(6)                                              SUMOUT
      WRITE (6,1130)  SV(12)                                            SUMOUT
      WRITE (6,1135)  SV(8)                                             SUMOUT
      WRITE (6,1140)  SV(9)                                             SUMOUT
      WRITE (6,1145)  SV(10)                                            SUMOUT
      WRITE (6,1150) SQ(14,4)                                           SUMOUT
      WRITE (6,1155) SQ(14,5)                                           SUMOUT
      WRITE (6,1160) SQ(15,3)                                           SUMOUT
      WRITE (6,1165) SQ(15,4)                                           SUMOUT
      WRITE (6,1195)  SW(15)                                            SUMOUT
C                                                                       SUMOUT
C     FIXED SOLID STRAPON SYNTHESIS OPTION   ************************   SUMOUT
      IF(SQ(20,1).LE.0.) GO TO 900                                      SUMOUT 900
      WRITE (6,1200) SQ(20,1),SQ(22,1),SQ(20,5),SQ(21,5),SQ(21,1),      SUMOUT
     1 SQ(21,2) ,SQ(20,4),SQ(20,2),SQ(20,3),SQ(22,2)                    SUMOUT
C     FIXED SOLID STRAPON SYNTHESIS OPTION   ************************   SUMOUT
C                                                                       SUMOUT
  900 CONTINUE                                                          SUMOUT
      WRITE (6,1000)                                                    SUMOUT
      WRITE (6,3000) SQ(36,1)                                           SUMOUT
      WRITE (6,3010) SQ(36,2)                                           SUMOUT
      WRITE (6,3015) SQ(36,3)                                           SUMOUT
      WRITE (6,3020) SQ(36,4)                                           SUMOUT
      WRITE (6,3025) SQ(36,5)                                           SUMOUT
      WRITE (6,3045) SQ(37,1)                                           SUMOUT
      WRITE (6,3046) SQ(37,2)                                           SUMOUT
```

```
      WRITE (6,3050)                                   SUMOUT
      IFLY = SQ(19,5)                                  SUMOUT
      GO TO (910,920,930,940,950),IFLY                 SUMOUT
  910 WRITE (6,3055) SW(9),SQ(10,2)                    SUMOUT
      GO TO 960                                        SUMOUT
  920 WRITE (6,3060) SV(12)                            SUMOUT
      GO TO 960                                        SUMOUT
  930 WRITE (6,3065) SQ(10,3)                          SUMOUT
      GO TO 960                                        SUMOUT
  940 WRITE (6,3070) SW(10),SQ(12,1)                   SUMOUT
      GO TO 960                                        SUMOUT
  950 WRITE (6,3075) SV(27)                            SUMOUT
      WRITE (6,3080) TLAT                              SUMOUT
      WRITE(6,3085) TLNG                               SUMOUT
      WRITE (6,3090) PZ(5)                             SUMOUT
      WRITE (6,3095) PZ(3)                             SUMOUT
      WRITE (6,3100) PZ(1)                             SUMOUT
      WRITE (6,3105) PZ(2)                             SUMOUT
      WRITE (6,3110) PZ(4)                             SUMOUT
  960 IFUL = SQ(32,1)                                  SUMOUT
      GO TO (965,970,975),IFUL                         SUMOUT
  965 WRITE (6,3115) SW(15)                            SUMOUT
      WRITE (6,3120) SW(12)                            SUMOUT
      WRITE (6,3125) SW(11)                            SUMOUT
      WRITE (6,3130) SW(14)                            SUMOUT
      WRITE (6,3135) SQ(10,3)                          SUMOUT
      WRITE (6,3140) SQ(32,4)                          SUMOUT
      W2 = WABFUB - SQ(32,4)                           SUMOUT
      WRITE (6,3145) W2                                SUMOUT
      WRITE (6,3150) WABFUB                            SUMOUT
      WRITE (6,3155) SCB(214)                          SUMOUT
      GO TO 980                                        SUMOUT
  970 WRITE (6,3160) SW(15)                            SUMOUT
      WRITE (6,3175) SQ(32,5)                          SUMOUT
      WRITE (6,3165) SQ(32,2)                          SUMOUT
      WRITE (6,3180) SQ(33,1)                          SUMOUT
      WRITE (6,3215) SQ(35,2)                          SUMOUT
      WRITE (6,3190) SQ(33,4)                          SUMOUT
      WRITE (6,3195) SQ(34,2)                          SUMOUT
      WRITE (6,3200) SQ(34,5)                          SUMOUT
      WRITE (6,3205) SQ(35,5)                          SUMOUT
      WRITE (6,3220) SQ(35,3)                          SUMOUT
      WRITE (6,3170) SQ(32,3)                          SUMOUT
      WRITE (6,3185) SQ(33,2)                          SUMOUT
      WRITE (6,3225) SQ(35,4)                          SUMOUT
      WRITE (6,3135) SQ(10,3)                          SUMOUT
      WRITE (6,3210) SQ(32,4)                          SUMOUT
      WRITE (6,3150) WABFUB                            SUMOUT
      WRITE (6,3155) SCB(214)                          SUMOUT
      GO TO 980                                        SUMOUT
  975 WRITE (6,3230) SW(15)                            SUMOUT
      WRITE (6,3175) SQ(32,5)                          SUMOUT
      WRITE (6,3235) SQ(33,3)                          SUMOUT
      WRITE (6,3245) SQ(34,1)                          SUMOUT
      WRITE (6,3255) SQ(34,4)                          SUMOUT
      WRITE (6,3180) SQ(33,1)                          SUMOUT
      WRITE (6,3215) SQ(35,2)                          SUMOUT
      WRITE (6,3190) SQ(33,4)                          SUMOUT
      WRITE (6,3195) SQ(34,2)                          SUMOUT
      WRITE (6,3200) SQ(34,5)                          SUMOUT
      WRITE (6,3205) SQ(35,5)                          SUMOUT
      WRITE (6,3220) SQ(35,3)                          SUMOUT
      WRITE (6,3240) SQ(33,5)                          SUMOUT
      WRITE (6,3250) SQ(34,3)                          SUMOUT
      WRITE (6,3260) SQ(35,1)                          SUMOUT
      WRITE (6,3185) SQ(33,2)                          SUMOUT
      WRITE (6,3225) SQ(35,4)                          SUMOUT
      WRITE (6,3135) SQ(10,3)                          SUMOUT
      WRITE (6,3210) SQ(32,4)                          SUMOUT
      WRITE (6,3150) WABFUB                            SUMOUT
      WRITE (6,3155) SCB(214)                          SUMOUT
```

```
      980 CONTINUE                                                       SUMOUT
          RETURN                                                         SUMOUT
C                                                                        SUMOUT
     1000 FORMAT(1H1,25X,31HSPACE SHUTTLE SYNTHESIS SUMMARY)              SUMOUT
     1005 FORMAT (37X,7HBOOSTER11X,7HORBITER11X,7HVEHICLE/)               SUMOUT
     1010 FORMAT (12H WEIGHT (LB))                                        SUMOUT
     1011 FORMAT(32H     PROPELLANT, ASCENT LESS FPRF12.0, F18.0)         SUMOUT
     1012 FORMAT (27H     PROPELLANT,ASCENT FPR,F17.0,F18.0)              SUMOUT
     1015 FORMAT(30H     PROPELLANT, ASCENT TOTAL F14.0,2F18.0)           SUMOUT
     1020 FORMAT (36H     PROPELLANT, ORBIT MANEUVER     8X,2F18.0)       SUMOUT
     1022 FORMAT (26H     FLYBACK FUEL              3F18.0)               SUMOUT
     1025 FORMAT (26H     PROPELLANT, TOTAL         3F18.0)               SUMOUT
     1030 FORMAT (26H     PAYLOAD                   18X,2F18.0)           SUMOUT
     1035 FORMAT(26H     STRUCTURE                3F18.0)                 SUMOUT
     1040 FORMAT (26H     OTHER                     3F18.0)               SUMOUT
     1041 FORMAT (26H     CONTINGENCY               3F18.0)               SUMOUT
     1045 FORMAT (26H     TOTAL                     3F18.0)               SUMOUT
     1050 FORMAT (13H0VOLUME (FT3))                                       SUMOUT
     1055 FORMAT (26H     FUEL TANK                 3F18.0)               SUMOUT
     1060 FORMAT (26H     OXIDIZER TANK             3F18.0)               SUMOUT
     1065 FORMAT (26H     PROPELLANT TANKS          3F18.0)               SUMOUT
     1070 FORMAT (26H     PAYLOAD                   18X,2F18.0)           SUMOUT
     1075 FORMAT (26H     OTHER                     3F18.0)               SUMOUT
     1077 FORMAT (26H     TOTAL                     3F18.0)               SUMOUT
     1080 FORMAT (9H GEOMETRY)                                            SUMOUT
     1085 FORMAT (26H     LENGTH (FT)               3F18.1)               SUMOUT
     1090 FORMAT (27H     BODY WETTED AREA (FT2)       F17.1,2F18.1)      SUMOUT
     1095 FORMAT (29H     BODY PLANFORM AREA (FT2)     F15.1,2F18.1)      SUMOUT
     1096 FORMAT (33H     THEORETICAL WING AREA (FT2) F11.1,F18.1)        SUMOUT
     1100 FORMAT (23H     WING LOADING (PSF) F21.1,F18.1)                 SUMOUT
     1105 FORMAT (11H PROPULSION)                                         SUMOUT
     1110 FORMAT (26H     THRUST-TO-WEIGHT          18X,2F18.5)           SUMOUT
     1111 FORMAT (26H     NO. OF ENGINES            2F18.0)               SUMOUT
     1112 FORMAT (26H    SL THRUST/ENG NOM     ,2(E10.0,8X), F9.0)        SUMOUT
     1113 FORMAT (26H    VAC THRUST/ENG NOM    ,2(F10.0,8X), F9.0)        SUMOUT
     1115 FORMAT (26H    SL ISP NOM            ,54X,F12.1)                SUMOUT
     1120 FORMAT (26H    VAC ISP NOM           ,54X,F12.1)                SUMOUT
     1125 FORMAT (11H0TRAJECTORY )                                        SUMOUT
     1126 FORMAT (26H     MASS RATIO                2F18.5)               SUMOUT
     1127 FORMAT (35H     MAXIMUM DYNAMIC PRESSURE (PSF)27X,F18.1)        SUMOUT
     1128 FORMAT(34H     CHARACTERISTIC VELOCITY (FPS)3X,F7.0,2F18.0)     SUMOUT
     1130 FORMAT (35H     STAGING DYNAMIC PRESSURE (PSF)27X,2F18.0)       SUMOUT
     1135 FORMAT (38H     STAGING VELOCITY (RELATIVE) (FPS)24X,F18.0)     SUMOUT
     1140 FORMAT (35H     STAGING ALTITUDE (FT)        27X,F18.0)         SUMOUT
     1145 FORMAT (47H     STAGING FLIGHT PATH ANGLE (RELATIVE) (DEG)15X,  SUMOUT
         1F18.3)                                                          SUMOUT
     1150 FORMAT (40H     INJECTION VELOCITY (INERTIAL) (FPS)4X,F18.0)    SUMOUT
     1155 FORMAT (35H     INJECTION ALTITUDE (FT)       9X,2F18.0)        SUMOUT
     1160 FORMAT (49H     INJECTION FLIGHT PATH ANGLE (INERTIAL) (DEG)F13.3) SUMOUT
     1165 FORMAT (35H     INJECTION INCLINATION (DEG)    9X,2F18.2)       SUMOUT
     1170 FORMAT (26H0     IN ORBIT                  18X,2F18.0)          SUMOUT
     1175 FORMAT (26H     RETURN CONDITION          2F18.0)               SUMOUT
     1180 FORMAT (26H     ENTRY                     2F18.0)               SUMOUT
     1185 FORMAT (26H     LANDING                   2F18.0)               SUMOUT
     1195 FORMAT (26H0FLYBACK RANGE (N MI)           F18.1)               SUMOUT
     1200 FORMAT (22H0SOLID AUGMENTATION    ,F4.0,7H SOLIDS,2X,15HPROPELLANT  SUMOUT
         1WT.= F8.0,2X,10HINERT WT.= F8.0/                                SUMOUT
         2                               12X,11HTOTAL WT. =F9.0,5X,11HEXIT A SUMOUT
         3REA = F6.0,5X,11HBURN TIME =F5.0/                               SUMOUT
         4                               12X,5HISP =F7 2, 5X,16HVAC THRUST/ SUMOUT
         5ENG= F8.0,4H + (,F7.0,9H + TIME )  / 12X,15HSL THRUST/ENG = F9.0)  SUMOUT
     2112 FORMAT (26H    SL THRUST/ENG NOM     ,F10.0,8X,F10.0,8X,F9.0)   SUMOUT
     2113 FORMAT (26H    VAC THRUST/ENG NOM    ,F10.0,8X,F10.0,8X,F9.0)   SUMOUT
     2115 FORMAT (26H    SL ISP NOM            ,36X,F12.1)                SUMOUT
     2120 FORMAT (26H    VAC ISP NOM           ,36X,F12.1)                SUMOUT
     3000 FORMAT (34H0DRAG LOSS AT STAGING             = F10.2)           SUMOUT
     3010 FORMAT (34H GRAVITY LOSS AT STAGING          = F10.2)           SUMOUT
     3015 FORMAT (34H DRAG LOSS AT INJECTION           = F10.2)           SUMOUT
     3020 FORMAT (34H GRAVITY LOSS AT INJECTION        = F10.2)           SUMOUT
     3025 FORMAT (34H MISALIGNMENT LOSS AT INJECTION = F10.2)             SUMOUT
     3045 FORMAT (18H BOOSTER BURN TIME,15X,1H= F7.2)                     CKOUT
     3046 FORMAT (18H ORBITER BURN TIME,15X,1H= F7.2)                     CKOUT
     3050 FORMAT ( 9H0FLYBACK )                                           SUMOUT
     3055 FORMAT (24H DETAILED RANGE, CLVG = E16.7,5X,12HBANK ANGLE = F7.2 ) SUMOUT
     3060 FORMAT (27H F(Q STAGE), Q AT STAGING = F10.3)                   SUMOUT
```

```
      3065 FORMAT (17H CONSTANT, DRNG = F10.3)                                  SUMOUT
      3070 FORMAT (22H IIP, CENTRAL ANGLE = F8.2,5X,5H IIP= F8.2)               SUMOUT
      3075 FORMAT (38H NUMERICAL INTEGRATION, MAX. LOADING = F7.3)              SUMOUT
      3080 FORMAT (5X,6HTLAT = F8.3)                                            SUMOUT
      3085 FORMAT (5X,6HTLNG = F8.3)                                            SUMOUT
      3090 FORMAT (5X15HSURFACE RANGE = F9.3)                                   SUMOUT
      3095 FORMAT (5X10HALTITUDE = F10.1)                                       SUMOUT
      3100 FORMAT (5X 10HLATITUDE = F8.3)                                       SUMOUT
      3105 FORMAT (5X,11HLONGITUDE = F8.3)                                      SUMOUT
      3110 FORMAT (5X,15HAZIMUTH DIFF. = F8.3)                                  SUMOUT
      3115 FORMAT (16HOCRUISE, RANGE = F8.2)                                    SUMOUT
      3120 FORMAT (5X,5HSFC = F9.4)                                             SUMOUT
      3125 FORMAT (5X,5HL/D = F9.4)                                             SUMOUT
      3130 FORMAT (5X,17HCRUISE VELOCITY = F9.2)                                SUMOUT
      3135 FORMAT (5X,6HDRNG = F9.2)                                            SUMOUT
      3140 FORMAT (5X,7HWFLYX = F9.2)                                           SUMOUT
      3145 FORMAT (5X,13HCRUISE FUEL = F10.2)                                   SUMOUT
      3150 FORMAT (5X,14HFLYBACK FUEL = F10.2)                                  SUMOUT
      3155 FORMAT (5X,10H C(214) =   F9.5)                                      SUMOUT
      3160 FORMAT (27HOPERCENTAGE WEIGHT, RANGE = F8.2)                         SUMOUT
      3165 FORMAT (5X,5HCA = F9.4)                                              SUMOUT
      3170 FORMAT (5X,5HCB = F9.4)                                              SUMOUT
      3175 FORMAT (5X,5HRT = F9.2)                                              SUMOUT
      3180 FORMAT (5X,5HR1 = F9.2)                                              SUMOUT
      3185 FORMAT (5X,5HR3 = F9.2)                                              SUMOUT
      3190 FORMAT (5X,5HSFC2=F9.4)                                              SUMOUT
      3195 FORMAT (5X,5HALD2=F9.4)                                              SUMOUT
      3200 FORMAT (5X,6HVFLY2=F9.4)                                             SUMOUT
      3205 FORMAT (5X,5HR2 = F9.2)                                              SUMOUT
      3210 FORMAT (5X,6HWFLYX= F9.2)                                            SUMOUT
      3215 FORMAT (5X20HIDLE DESCENT FUEL = F9.2)                               SUMOUT
      3220 FORMAT (5X20HCRUISE FUEL WEIGHT = F10.2)                             SUMOUT
      3225 FORMAT (5X,27HFINAL DESCENT FUEL WEIGHT = F9.2)                      SUMOUT
      3230 FORMAT (17HOBREGUET, RANGE = F9.2)                                   SUMOUT
      3235 FORMAT (5X,7HSFC1 = F9.4)                                            SUMOUT
      3240 FORMAT (5X,7HSFC3 = F9.4)                                            SUMOUT
      3245 FORMAT (5X,7HALD1 = F9.4)                                            SUMOUT
      3250 FORMAT (5X,7HALD3 = F9.4)                                            SUMOUT
      3255 FORMAT (5X,7HVFLY1= F9.4)                                            SUMOUT
      3260 FORMAT (5X,7HVFLY3= F9.4)                                            SUMOUT
           END                                                                  SUMOUT
```

1554

SUBRØUTINE TAMPER

1555

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
BTTOT		I	Total booster weight flow	/TAMP /(	5)	SUMOUT	I	BTTOT
						TAMPER	I	BTTOT
						WTVOL	O	BTTOT
BWDOT1		W	Booster weight flow	/TAMPER/(*	)	TAMPER	W	BWDOT1
IFIRE		W	Integer value of fire flag	/TAMPER/(*	)	TAMPER	W	IFIRE
LBODY		I	Body length	/VOLCAL/(	8)	PROTHR	I	LBODY
						TAMPER	I	LBODY
						WTSCH	M	LBODY
OTTOT		I	Total orbiter thrust	/TAMP /(	2)	SUMOUT	I	OTTOT
						TAMPER	I	OTTOT
						WTVOL	M	OTTOT
SBODY		I	Total body wetted area	/CINPUT/(	386)	PROTHR	I	SBODY
						TAMPER	I	SBODY
						WTSCH	M	SBODY
SBODYB		O	Total body wetted area- booster	/SUMVW /(	13)	SUMOUT	I	SBODYB
						TAMPER	O	SBODYB
SBODYO		O	Total body wetted area- orbiter	/SUMVW /(	26)	SUMOUT	I	SBODYO
						TAMPER	O	SBODYO
SCB		I	Working name for input c-array booster scaling coefficients	/ORBINY/(	144)	FLYBKP	M	SCB
						STORE	M	SCB
						SUMOUT	I	SCB
						TAMPER	I	SCB
						THRUST	M	SCB
						VEHDF	I	SCB
						WTVOL	M	SCB
SE		M	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SPLAN		I	Body planform area	/VOLCAL/(	14)	PROTHR	I	SPLAN
						TAMPER	I	SPLAN
						WTSCH	O	SPLAN
						WTVOL	I	SPLAN
SPLANB		O	Booster body planform area	/SUMVW /(	14)	SUMOUT	I	SPLANB
						TAMPER	O	SPLANB
SPLANO		O	Orbiter body planform area	/SUMVW /(	27)	SUMOUT	I	SPLANO
						TAMPER	O	SPLANO
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ

1556

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SV		M	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TAMPER		E	Subroutine to interface weight and volume overlay with trajectory program overlay	/TAMPER/(	)	TAMPER	E	TAMPER
						WTVOL	S	TAMPER
TB27		I	Stored booster value of Isp(I)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		I	Stored booster value of mr(I)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TB36		I	Stored booster value of nengs	/ORBINY/(	60)	STORE	M	TB36
						SUMOUT	I	TB36
						TAMPER	I	TB36
						THRUST	I	TB36
						WTVOL	I	TB36
TB45		I	Stored booster value of swing	/ORBINY/(	69)	STORE	M	TB45
						SUMOUT	I	TB45
						TAMPER	I	TB45
TO34		I	Stored orbiter value of mr(I)	/ORBINX/(	53)	ITER8	M	TO34
						SSSP	O	TO34
						STORE	M	TO34
						TAMPER	I	TO34
						VEHDF	O	TO34
						WTVOL	M	TO34
TO36		I	Stored orbiter value of nengs	/ORBINX/(	60)	STORE	M	TO36
						SUMOUT	I	TO36
						TAMPER	I	TO36
						THRUST	I	TO36
						WTVOL	I	TO36
TO45		I	Stored orbiter value of swing	/ORBINX/(	69)	STORE	M	TO45
						SUMOUT	I	TO45
						TAMPER	I	TO45
T2		W	Main engine burn time	/TAMPER/(	)	TAMPER	W	T2
VABFUB		O	Volume of booster propellant tanks	/SUMVW /(	11)	TAMPER	O	VABFUB
VBODY		I	Total body volume	/CINPUT/(	391)	PRINTV	M	VBODY
						SOLVE	M	VBODY
						STORE	M	VBODY
						TAMPER	I	VBODY
						WTSCH	M	VBODY
						WTVOL	I	VBODY
VBODYB		O	Booster body volume	/SUMVW /(	10)	SUMOUT	I	VBODYB
						TAMPER	O	VBODYB
VBODYO		O	Orbiter body volume	/SUMVW /(	24)	SUMOUT	I	VBODYO
						TAMPER	O	VBODYO

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
VCARGO		I	Volume of cargo bay	/VOLCAL/(	27)	PRINTV TAMPER WTSCH	I I M	VCARGO VCARGO VCARGO
VCARO		O	Volume of cargo	/SUMVW /(	22)	SUMOUT TAMPER	I O	VCARO VCARO
VFUTK		I	Total volume of fuel tank	/VOLCAL/(	29)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK VFUTK VFUTK VFUTK
VFUTKB		O	Total volume of fuel tank - booster	/SUMVW /(	7)	SUMOUT TAMPER	I O	VFUTKB VFUTKB
VFUTKO		O	Total volume of fuel tank - orbiter	/SUMVW /(	20)	SUMOUT TAMPER	I O	VFUTKO VFUTKO
VFUTK2		I	Total volume of secondary fuel tank	/VOLCAL/(	30)	PRINTV STORE TAMPER WTSCH	I M I M	VFUTK2 VFUTK2 VFUTK2 VFUTK2
VOTHB		O	Misc. Booster volume	/SUMVW /(	9)	SUMOUT TAMPER	I O	VOTHB VOTHB
VOTHO		O	Misc. Orbiter volume	/SUMVW /(	23)	SUMOUT TAMPER	I O	VOTHO VOTHO
VOXTK		I	Total volume of oxidizer tank	/VOLCAL/(	34)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK VOXTK VOXTK VOXTK
VOXTKB		O	Booster oxidizer tank volume	/SUMVW /(	8)	SUMOUT TAMPER	I O	VOXTKB VOXTKB
VOXTKO		O	Orbiter oxidizer tank volume	/SUMVW /(	21)	SUMOUT TAMPER	I O	VOXTKO VOXTKO
VOXTK2		I	Total volume of secondary oxidizer tank	/VOLCAL/(	35)	PRINTV STORE TAMPER WTSCH	I M I M	VOXTK2 VOXTK2 VOXTK2 VOXTK2
WABFU		I	Weight of jp fuel	/WTCALC/(	3)	PRINTW PRWTSM TAMPER WTSCH	I M I M	WABFU WABFU WABFU WABFU
WABFUB		M	Booster flyback fuel required	/SUMVW /(	2)	SUMOUT TAMPER	I M	WABFUB WABFUB
WABFUO		M	Orbiter flyback fuel required	/SUMVW /(	19)	SUMOUT TAMPER	I M	WABFUO WABFUO
WBOO		I	Booster gross weight	/JUMPY /(	3)	PRINTW TAMPER WTVOL	I I O	WBOO WBOO WBOO
WCONT		I	Contingency and growth weight	/WTCALC/(	16)	PRINTW TAMPER WTSCH	I I M	WCONT WCONT WCONT
WCONTB		O	Contingency and growth weight-boostr	/SUMVW /(	38)	SUMOUT TAMPER	I O	WCONTB WCONTB
WCONTO		O	Contingency and growth weight-orbitr	/SUMVW /(	37)	SUMOUT TAMPER	I O	WCONTO WCONTO
WDOTT1		W	Weight flow	/TAMPER/(*	)	TAMPER	W	WDOTT1
WDRY		I	Stage dry weight	/WTCALC/(	24)	PRINTW TAMPER WTSCH	I I M	WDRY WDRY WDRY
WDRYB		O	Stage dry weight - booster	/SUMVW /(	4)	SUMOUT TAMPER	I O	WDRYB WDRYB
WDRYO		O	Stage dry weight -orbiter	/SUMVW /(	17)	SUMOUT TAMPER	I O	WDRYO WDRYO
WENTRB		O	Booster entry weight	/SUMVW /(	34)	SUMOUT TAMPER	I O	WENTRB WENTRB

1552

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WENTRO		O	Orbiter entry weight	/SUMVW /(	33)	SUMOUT TAMPER	I O	WENTRO WENTRO
WENTRY		I	Entry weight	/EMS /(	3)	PRWTSM TAMPER	O I	WENTRY WENTRY
WETURN		I	Weight at return point	/EMS /(	2)	PRWTSM TAMPER	O I	WETURN WETURN
WFUOX		I	Weight of main and secondary propellant	/WTCALC/(	46)	PRINTW TAMPER WTSCH WTVOL	I I M I	WFUOX WFUOX WFUOX WFUOX
WFUOXB		M	Propellant wt. Less fpr -booster	/SUMVW /(	3)	SUMOUT TAMPER	I M	WFUOXB WFUOXB
WFUOXO		I	Propellant wt. Less fpr -orbiter	/TAMP /(	3)	SUMOUT TAMPER WTVOL	I I M	WFUOXO WFUOXO WFUOXO
WFURES		I	Fuel reserve	/WTCALC/(	47)	PRINTW STORE TAMPER WTSCH	I M I M	WFURES WFURES WFURES WFURES
WFU2		I	Weight of secondary fuel	/WTCALC/(	34)	PRINTW TAMPER WTSCH	I I M	WFU2 WFU2 WFU2
WGROSB		M	Booster gross weight	/SUMVW /(	5)	SUMOUT TAMPER	I M	WGROSB WGROSB
WGROSS		I	Gross lift-off weight	/CINPUT/(	392)	PRINTW PRWTSM SOLVE STORE TAMPER WTSCH WTVOL	I I M M I M I	WGROSS WGROSS WGROSS WGROSS WGROSS WGROSS WGROSS
WJET		I	Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA PRWTSM STORE TAMPER WTSCH	I M O I M	WJET WJET WJET WJET WJET
WLAND		I	Landing weight	/EMS /(	4)	PRWTSM TAMPER	O I	WLAND WLAND
WLANDB		O	Landing weight - booster	/SUMVW /(	35)	SUMOUT TAMPER	I O	WLANDB WLANDB
WLANDO		O	Landing weight - orbiter	/SUMVW /(	36)	SUMOUT TAMPER	I O	WLANDO WLANDO
WOP		O	Orbit maneuvering propellant weight	/SUMVW /(	1)	SUMOUT TAMPER	I O	WOP WOP
WORBIT		I	Weight in orbit	/EMS /(	1)	PRWTSM TAMPER	O I	WORBIT WORBIT
WORBTB		O	In-orbit weight - booster	/SUMVW /(	30)	TAMPER	O	WORBTB
WORBTO		O	In-orbit weight - orbiter	/SUMVW /(	29)	SUMOUT TAMPER	I O	WORBTO WORBTO
WOTHB		O	Misc. Weight - booster	/SUMVW /(	6)	SUMOUT TAMPER	I O	WOTHB WOTHB
WOTHO		O	Misc. Weight - orbiter	/SUMVW /(	18)	SUMOUT TAMPER	I O	WOTHO WOTHO
WOVERS		I	Wing loading	/WTCALC/(	77)	PROTHR TAMPER WTSCH	I I M	WOVERS WOVERS WOVERS
WOVRSB		O	Booster wing loading	/SUMVW /(	15)	SUMOUT TAMPER	I O	WOVRSB WOVRSB
WOVRSO		O	Orbiter wing loading	/SUMVW /(	28)	SUMOUT TAMPER	I O	WOVRSO WOVRSO

1557

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WOXRES		I	Oxidizer reserve	/WTCALC/(	89)	PRINTW STORE TAMPER WTSCH	I M I M	WOXRES WOXRES WOXRES WOXRES
WOX2		I	Secondary oxidizer weight	/WTCALC/(	84)	PRINTW TAMPER WTSCH	I I M	WOX2 WOX2 WOX2
WPAYL		I	Payload weight	/WTCALC/(	97)	PRINTW TAMPER WTSCH	I I M	WPAYL WPAYL WPAYL
WPAYLO		M	Payload weight	/SUMVW /(	16)	SSSP SUMOUT TAMPER	I I M	WPAYLO WPAYLO WPAYLO
WRTRNB		O	Entry weight- booster	/SUMVW /(	32)	SUMOUT TAMPER	I O	WRTRNB WRTRNB
WRTRNO		O	Entry weight- orbiter	/SUMVW /(	31)	SUMOUT TAMPER	I O	WRTRNO WRTRNO
WWAIT		I	Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA PRWTSM STORE TAMPER WTSCH	I I M I M	WWAIT WWAIT WWAIT WWAIT WWAIT

```
      SUBROUTINE TAMPER (NCALL)                                         TAMPER
C*****************************************************************     TAMPER
C           PRIMARY DATA INTERFACE BETWEEN WTVOL AND                    TAMPER
C           TRAJECTORY PROBRAM. THIS SUBROUTINE SETS                    TAMPER
C           TRAJ PROG WORKING VARIABLES AND STORES                      TAMPER
C           SUMMARY WEIGHT ADN VOLUME DATA                              TAMPER
C*****************************************************************     TAMPER
      COMMON / JUMPY / JUMP, WBIG, WBOO                                 TAMPER
      COMMON/TAMP/WGROSO,OTTOT,WFUOXO,TBTO,BTTOT                        TAMPER
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      COMMON/WTCALC/   ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                    CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,   ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27, ORBINX
     3 TO28,TO29,TO30,TO31,TO32,TO33,TO34,TO35,TO36,TO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                         CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,    UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,    UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                    CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,   ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27, ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                               UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,    UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,     UH
     8TB83,TB84                                                         UH
      REAL LBODYO,LBODYB                                                SUMVW
      COMMON/SUMVW/WOP,          WABFUB,WFUDXB,WDRYB,WGROSB,WOTHB,      SUMVW
     1 VFUTKB,VOXTKB,VOTHB,VBODYB,VABFUB,                               SUMVW
     2 LBODYB,SBODYB,SPLANB,WOVRSB,                                     SUMVW
     1                 WPAYLO,WDRYO,WOTHO,  WABFUO,                     SUMVW
     2 VFUTKO,VOXTKO,VCARO,VOTHO,VBODYO,                                SUMVW
     3 LBODYO,SBODYO,SPLANO,WOVRSO                                      SUMVW
     4 ,WORBTO,WORBTB,WRTRNO,WRTRNB,WENTRO,WENTRB,WLANDB,WLANDO         SUMVW
     5 , WCONTO,WCONTB                                                  SUMVW
```

```
      REAL MUB, MUO, ISPB, ISPO, IDVEL,WNB,NO                                  SIZING
      COMMON /SIZING/                                                          SIZING
C         PHASE II SIZING PARAMERERS                                           SIZING
     *TZ,      VV(3),      OP(14),   EROR,     PZ(5),     VQ,       SW(20),    SIZING
     *SV(28),  SQ(37,5),   SE(11),   TLAT,     TLNG,                           SIZING
C        PHASE I SIZING PARAMERERS                                             SIZING
     *WBO,     WLOO,       DWEB,     DWEO,     TOLWT,     WPB,      TWRAT2,    SIZING
     *BK1,     BK2,        BK3,      BK4,      ISIZE,     TRAFLG,   TWRATO,    SIZING
     *OK1,     OK2,        OK3,      OK4,      PRFLG,     IPASS,    IPSMAX,    SIZING
     *AEXIT,   TVACO,      NO,       WFO,      IDVEL,     ISPO,     ISPB,      SIZING
     *XPL,     TVACB,    WNB,        WEO,      WEB,       WO,       WLO,       SIZING
     *DVO,      DVB,       MUB,       MUO,      VSTG,      WPO                 SIZING
     *,JTYP     BECO       BSTG      ORBI     ITNBW , ITNOW   ,                SIZING
     * SVOPSQ  , SVOCON   ,IHUNT   ,IOPSTG  ,ISZO(19)                          UH
      COMMON/EMS/WORBIT,WETURN,WENTRY,WLAND                                    TAMPER
      COMMON/OS/OSPLAN                                                         TAMPER
C                                                                              TAMPER
      IFIRE = SE(2)                                                            TAMPER
      VV(1)  = WGROSS                                                          TAMPER
      SV(7) = OWSAVE(1)                                                        CKOUT
      SV(4) = OWSAVE(4)                                                        CKOUT
      SV(6) = TO34(3)                                                          CKOUT
      SQ(19,4) = WJET(5)                                                       CKOUT
C                                                                              CKOUT
C         PROPULSION REFERENCE AREA (TOTAL)                                    CKOUT
C                                                                              CKOUT
      QP(11) = TB36 * SCB(218)                                                 CKOUT
      QP(12) = TO36 * SCO(218)                                                 CKOUT
C                                                                              TAMPER
C         AERODYNAMIC REFERENCE AREAS                                          TAMPER
C                                                                              TAMPER
C     QP(6) =  AERO REF AREA WITH SOLID MOTORS, BOOSTER,AND ORBITER            TAMPER
      QP(7) = TB45                                                             TAMPER
      QP(8) = TO45                                                             TAMPER
C                                                                              TAMPER
      QP(10) = SCO(103)                                                        CKOUT
      QP(9) = VV(1) - WFUOX                                                    CKOUT
C                                                                              TAMPER
C         ORBITER VACUUM THRUST AND ISP                                        TAMPER
C                                                                              TAMPER
      QP(2) = OTTOT * SE(8)                                                    TAMPER
      QP(4) = SE(7)                                                            TAMPER
C                                                                              TAMPER
C         BOOSTER WEIGHT FLOW                                                  TAMPER
C                                                                              TAMPER
      BWDOT1 = BTTOT * SE(9)/ SE(1)                                            TAMPER
      SE(11) = 0.                                                              TAMPER
C                                                                              TAMPER
C         TEST FOR PARALLEL BURN                                               TAMPER
C                                                                              TAMPER
      GO TO (5,9) , IFIRE                                                      TAMPER  5 9
C                                                                              TAMPER
C         SIMULTANEOUS FIRE OF BOOSTER AND ORBITER                             TAMPER
    5 CONTINUE                                                                 TAMPER
C                                                                              TAMPER
C         VACUUM THRUST AND ISP                                                TAMPER
C                                                                              TAMPER
      QP(1) = OTTOT* SE(8) + BOTTOT* SE(9)                                     TAMPER
      OWDOT1 = OTTOT * SE(8)/ SE(7)                                            TAMPER
      WDOTT1 = OWDOT1 + BDOT1                                                  TAMPER
      QP(3)  = QP(1)/WDOTT1                                                    TAMPER
C                                                                              CKOUT
C         CALCULATE STAGE BURN TIMES                                           CKOUT
C                                                                              CKOUT
      SQ(37,1) = (WFUOXB + SQ(37,5) )* QP(4)/QP(2)                             CKOUT
      SQ(37,2) = (WFUOXO + SQ(37,4) )* QP(1)/QP(3)                             CKOUT
C                                                                              TAMPER
C         CHECK PROPELLANT CROSS FEED FLAG                                     TAMPER
C                                                                              TAMPER
      IF(SE(10).EQ.0.) GO TO 16                                                TAMPER  16
      T2 = ( WFUOX - BWDOT1 * QP(14))/ BWDOT2                                  TAMPER
      SE(11) = OWDOT1 +  QP(14) + OWDOT2 * T2                                  TAMPER
      QP(10) = QP(10) + SE(11)                                                 TAMPER
      QP(9)  =  QP(9)  - SE(11)                                                TAMPER
```

1562

```
    9 CONTINUE                                                         TAMPER
C     SEQUENTIAL FIRING OF BOOSTER AND ORBITER   OR                    TAMPER
C     TANDEM FIRING OF BOOSTER AND ORBITER                             TAMPER
C     THRUST AND ISP                                                   TAMPER
C                                                                      TAMPER
      QP(1)  =  BTTOT * SE(9)                                          TAMPER
      QP(3)  =  QP(1)/ BWDOT1                                          TAMPER
   16 CONTINUE                                                         TAMPER
      SV(11) = QP(1)/ VV(1)                                            CKOUT
      SV(13) =  SV(11) * TB27(2)/QP(3)                                 TAMPER
C                                                                      TAMPER
C          CHECK SOLID MOTOR OPTION FLAG                               TAMPER
C                                                                      TAMPER
      IF(SQ(20,1).LE.0.) GO TO 18                                      TAMPER  18
      VV(1) = VV(1) + SQ(21,5)                                         TAMPER
      SV(13) =(QP(1) * TB27(2)/QP(3) + SQ(21,4))/VV(1)                 TAMPER
      TZ = (QP(1) + SQ(22,4))/( BWDOT1 + SQ(21,3)/SQ(21,2))            CKOUT
   18 IF(NCALL.EQ.1) RETURN                                            TAMPER
      SV(14) = TB34(3)                                                 TAMPER
      SV(20) = WWAIT(6)                                                TAMPER
      SQ(4,1) = OWSAVE(4)-WPAYLO                                       OS
      SV(5) = OWSAVE(1)-OWSAVE(4)                                      OS
      VV(2) = QP(9) - SV(7)                                            CKOUT
      VV(3) = WFUOXB                                                   CKOUT
      SV(16) = SV(13) - 1.                                             CKOUT
      RETURN                                                           TAMPER
C******************************************************************    TAMPER
C                                                                      TAMPER
      ENTRY ORBSUM                                                     TAMPER
C     ORBITER SUMMARY WEIGHTS                                          TAMPER
C                                                                      TAMPER
      SQ(37,4) = WFURES + WOXRES                                       TAMPER
      WOP=WFU2(1)+WOX2(1)                                              TAMPER
      WPAYLO = WPAYL                                                   TAMPER
      WABFUO = ABS(WABFU)                                              TAMPER
      WCONTO = WCONT                                                   TAMPER
      WORYO = WDRY                                                     TAMPER
      WOTHO = WGROSS - WFUOXO - WABFUO - WPAYL - WDRY - SQ(37,4) - WCONT TAMPER
      WORBTO = WORBIT                                                  TAMPER
      WRTRNO = WETURN                                                  TAMPER
      WENTRO = WENTRY                                                  TAMPER
      WLANDO = WLAND                                                   TAMPER
C                                                                      TAMPER
C     ORBITER SUMMARY VOLUMES                                          TAMPER
C                                                                      TAMPER
      VFUTKO = VFUTK                                                   TAMPER
      VOXTKO = VOXTK                                                   TAMPER
      VCARO = VCARGO                                                   TAMPER
      VOTHO = VBODY - VCARGO - VOXTK - VFUTK                           TAMPER
      VBODYO = VBODY                                                   TAMPER
C                                                                      TAMPER
C      ORBITER SUMMARY GEOMETRY                                        TAMPER
C                                                                      TAMPER
      LBODYO = LBODY                                                   TAMPER
      SBODYO = SBODY                                                   TAMPER
      SPLANO = SPLAN                                                   TAMPER
      WOVRSO = WOVERS                                                  TAMPER
      RETURN                                                           TAMPER
C                                                                      TAMPER
      ENTRY BOOSUM                                                     TAMPER
C     BOOSTER SUMMARY WEIGHTS                                          TAMPER
C                                                                      TAMPER
      SQ(37,5) = WFURES + WOXRES                                       TAMPER
      WABFUB = ABS(WABFU)                                              TAMPER
      WFUOXB = WFUOX                                                   TAMPER
      WCONTB = WCONT                                                   TAMPER
      WDRYB = WDRY                                                     TAMPER
      WGROSB = WBOO                                                    TAMPER
      WOTHB = WGROSB - WFUOX - WABFUB - WDRY - SQ(37,5) - WCONT        TAMPER
      WORBTB = WORBIT                                                  TAMPER
      WRTRNB = WETURN                                                  TAMPER
      WENTRB = WENTRY                                                  TAMPER
      WLANDB = WLAND                                                   TAMPER
C                                                                      TAMPER
```

1563

```
C      BOOSTER SUMMARY VOLUMES                                   TAMPER
C                                                                TAMPER
      VFUTKB = VFUTK                                             TAMPER
      VOXTKB = VOXTK                                             TAMPER
      VOTHB = VBODY - VFUTK - VOXTK                              TAMPER
      VABFUB = VOXTK2 + VFUTK2                                   TAMPER
      VBODYB = VBODY                                             TAMPER
C                                                                TAMPER
C     BOOSTER SUMMARY GEOMETRY                                   TAMPER
C                                                                TAMPER
      LBODYB = LBODY                                             TAMPER
      SBODYB = SBODY                                             TAMPER
      SPLANB = SPLAN                                             TAMPER
      WOVRSB = WOVERS                                            TAMPER
      RETURN                                                     TAMPER
      END                                                        TAMPER
```

SUBRØUTINE TBL2D

1564

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
TBL2D		E	Two dimensional table look-up surboutine	/TBL2D /($	)	RANGE	S	TBL2D
						TBL2D	E	TBL2D

```
      SUBROUTINE TBL2D(X,Y,XTBL,YTBL,ZTBL,Z)                          TBL2D
C                                                                     TBL2D
C     2 DIMENSIONAL TABLE LOOK-UP                                     TBL2D
C                                                                     TBL2D
      DIMENSION XTBL(7), YTBL(7), ZTBL(7,7)                           TBL2D
      K = 0                                                           TBL2D
      DO 1 II=2,7                                                     TBL2D
      I=II                                                            TBL2D
      IF(X-XTBL(I)) 3,2,1                                             TBL2D
    1 CONTINUE                                                        TBL2D
      GO TO 3                                                         TBL2D
    2 K = 1                                                           TBL2D
      GO TO 4                                                         TBL2D
    3 SCAL1 = (X-XTBL(I-1))/(XTBL(I)-XTBL(I-1))                       TBL2D
    4 DO 5 JJ=2,7                                                     TBL2D
      J=JJ                                                            TBL2D
      IF(Y-YTBL(J)) 7,6,5                                             TBL2D
    5 CONTINUE                                                        TBL2D
      GO TO 7                                                         TBL2D
    6 SCAL2 = 1.                                                      TBL2D
      GO TO 8                                                         TBL2D
    7 SCAL2 = (Y-YTBL(J-1))/(YTBL(J)-YTBL(J-1))                       TBL2D
    8 IF(K-1) 9,10,9                                                  TBL2D
    9 Z1 = ZTBL(I-1,J-1) + SCAL1*(ZTBL(I,J-1) - ZTBL(I-1,J-1))        TBL2D
      Z2 = ZTBL(I-1,J) + SCAL1*(ZTBL(I,J) - ZTBL(I-1,J))              TBL2D
      Z = Z1 + SCAL2*(Z2 - Z1)                                        TBL2D
      RETURN                                                          TBL2D
   10 Z = ZTBL(I,J-1) + SCAL2*(ZTBL(I,J) - ZTBL(I,J-1))               TBL2D
C                                                                     TBL2D
      RETURN                                                          TBL2D
      END                                                             TBL2D
```

SUBRØUTINE
THRUP

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
LOCI		I	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the initial value of the independent variable of the corresponding table. A zero entry indicates table not input.	/TABLE /(	1)	SPLICO	M	LOCI
						SPLICO	M	X
						SPLIZ	I	LOCI
						SPLIZ	I	X
						SPLYNE	I	LOCI
						SPLYNE	I	X
						THRUP	I	LOCI
						THRUP	O	X
X		O	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the initial value of the independent variable of the corresponding table. A zero entry indicates table not input.	/TABLE /(	1)	SPLICO	M	LOCI
						SPLICO	M	X
						SPLIZ	I	LOCI
						SPLIZ	I	X
						SPLYNE	I	LOCI
						SPLYNE	I	X
						THRUP	I	LOCI
						THRUP	O	X
Y		O	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that indicates the last interval in which interpolation of the corresponding table occured.	/TABLE /(	701)	SPLICO	O	LOCL
						SPLICO	M	Y
						SPLIZ	M	LOCL
						SPLIZ	I	Y
						SPLYNE	M	LOCL
						SPLYNE	I	Y
						THRUP	O	Y
Z		O	A 50 word array that corresponds to tables 1 thru 50. Each entry is an integer that points to the last value of the independent variable of the corresponding table.	/TABLE /(	1401)	SPLICO	M	LOCF
						SPLICO	M	Z
						SPLIZ	I	LOCF
						SPLIZ	I	Z
						SPLYNE	I	LOCF
						SPLYNE	I	Z
						THRUP	O	Z

```
      SUBROUTINE THRUP(ITAB, F1, T2, F2)                                THRUP
C                                                                       THRUP
C     THIS SUBROUTINE SETS UP THE LINEAR THRUST                         THRUP
C     TABLE NUMBERED ITAB.  THE TABLE ENTRIES                           THRUP
C     F1, T2 AND F2 ARE DETERMINED BY THE SIZING                        THRUP
C     MODULE FOR USE IN THE TRAJECTORY MODULE.                          THRUP
      DIMENSION  LOCI(1)                                                THRUP
      EQUIVALENCE (LOCI,X)                                              THRUP
      COMMON/TABLE/ X(700),Y(700),Z(700)                                THRUP
C     FIND THE FIRST ENTRY OF ITAB.                                     THRUP
C                                                                       THRUP
      II = LOCI(ITAB)                                                   THRUP
C                                                                       THRUP
C     SET THE INITIAL TIME OF ITAB TO ZERO AND                          THRUP
C     ZERO OUT THE SECOND DERIVATIVES.                                  THRUP
C                                                                       THRUP
      X(II) = 0.                                                        THRUP
      Z(II) = 0.                                                        THRUP
      Z(II + 1) = 0.                                                    THRUP
C                                                                       THRUP
C     STORE THE FINAL TIME OF ITAB AND THE INITIAL                      THRUP
C     AND FINAL VALUES OF THRUST.                                       THRUP
C                                                                       THRUP
      X(II + 1) = T2                                                    THRUP
      Y(II) = F1                                                        THRUP
      Y(II + 1) = F2                                                    THRUP
      RETURN                                                            THRUP
      END                                                               THRUP
```

1570

SUBRØUTINE THRUST

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBR	CODE	VAR
FSLB		M	Booster sea level thrust (lb)	/TRUST /(	6)	THRUST	M	FSLB
FSLLO		M	Sea level lift-off thrust (lb)	/TRUST /(	7)	THRUST	M	FSLLO
FSLO		M	Orbiter sea level thrust (lb)	/TRUST /(	2)	THRUST	M	FSLO
FSLS		M	Solid motor sea level thrust (lb)	/TRUST /(	8)	THRUST	M	FSLS
FUNIT		W	Orbiter thrust per engine vac. (lb)	/THRUST/(*	)	THRUST	W	FUNIT
FVACB		M	Booster vacuum thrust (lb)	/TRUST /(	5)	THRUST WTSCH	M M	FVACB FVACB
FVACLO		M	Total vacuum lift-off thrust (lb)	/TRUST /(	3)	THRUST	M	FVACLO
FVACS		M	Solid motor total vacuum thrust (lb)	/TRUST /(	4)	THRUST	M	FVACS
MT		I	Curve number -thrust table	/ARCDAT/(	25)	FXDAT PROPB PROPIN THRUST	I I I I	MT MT MT MT
SCB		M	Working name for input c-array booster scaling coefficients	/ORBINY/(	144)	FLYBKP STORE SUMOUT TAMPER THRUST VEHDF WTVOL	M M I I M I M	SCB SCB SCB SCB SCB SCB SCB
SE		I	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP PRITVA SIZEMR SUMOUT TAMPER THRUST VEHDF WTSCH WTVOL	O I I I M I M I M	SE SE SE SE SE SE SE SE SE
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM FLYBKP ISPRAT PDBC PRITVA RANGE REU3 SIZE SIZEMR SIZIN STAU SUMOUT TAMPAR TAMPER THRUST TRTOSZ VEHDF WTVOL	M M I I I M O O M M I M O M M M M M	SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ SQ
SREF	S_{ref}	I	Aerodynamic reference area (FT^2)	/ARCDAT/(	1)	BNTG EQUA3 FNTG FXDAT FXDAT GEINP SDINP SIZIN SIZIN THRUST VT	I I I I O M I I M I I	ARCDA SREF ARCDA ARCDA IARCDA ARCDA ARCDA ARCDA SREF SREF SREF
TB36		I	Stored booster value of nengs	/ORBINY/(	60)	STORE SUMOUT TAMPER THRUST WTVOL	M I I I I	TB36 TB36 TB36 TB36 TB36
THRUST		E	Subroutine to determine booster, orbiter, and vehicle thrust	/THRUST/(S	)	DATAIN THRUST	S E	THRUST THRUST

1572

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T036		I	Stored orbiter value of nengs	/ORBINX/(	60)	STORE	M	T036
						SUMOUT	I	T036
						TAMPER	I	T036
						THRUST	I	T036
						WTVOL	I	T036

```
      SUBROUTINE THRUST                                                 THRUST
C                                                                       THRUST
C          SUBROUTINE TO CALCULATE BOOSTER, ORBITER, AND                THRUST
C          VEHICLE THRUST                                               THRUST
C                                                                       THRUST
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C           PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,       VV(3),      QP(14),    EROR,      PZ(5),     VQ,       SW(20), SIZING
     *SV(28),   SQ(37,5),   SE(11),    TLAT,      TLNG,                  SIZING
C          PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,      WLOO,       DWEB,      DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,        BK3,       BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,        OK3,       OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,      NO,        WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,     NNB,        WEO,       WEB,       WO,       WLO,    SIZING
     *DVO,       DVB,        MUB,       MUO,       VSTG,      WPO              SIZING
     *,JTYP,     BECO ,     BSTG ,     ORBI,     ITNBW ,  ITNOW  ,           SIZING
     * SVDPSQ  , SVDCON   ,IHUNT     ,IOPSTG   ,ISZD(19)                    UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),        ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                     CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,    ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,TO22,TO23,TO24,TO25,TO26,TO27, ORBINX
     3 TO28,TO29,TO30,TO31,TO32,TO33,TO34,TO35,TO36,TO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                          CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,      UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,      UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),        ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                     CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,    ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,TB22,TB23,TB24,TB25,TB26,TB27, ORBINY
     3 TB28,TB29,TB30,TB31,TB32,TB33,TB34,TB35,TB36,TB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,      UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,       UH
     8TB83,TB84                                                         UH
      COMMON/TRUST/                                                     CKOUT
     *FVACO    ,FSLO      ,FVACLO    ,FVACS,                            THRUST
     *FVACB    ,FSLB      ,FSLLO     ,FSLS                              THRUST
      COMMON/ARCDAT/                                                    ARCDAT
     *SREF        ,EJ          ,XISP        ,TMULT       ,DTNC        ,DTPI        , ARCDAT
     *IATM        ,IMODE       ,JAER        ,JPRO        ,QMAX        ,GMAX        , ARCDAT
     *XLMAX       ,HDMAX       ,GMDOT       ,ALFMAX      ,PHMAX       ,MAEA        , ARCDAT
     *MAEB        ,MAEC        ,MAED        ,MAEE        ,MAEF        ,MAEG        , ARCDAT
     *MT          ,MISP        ,MXCG        ,MZCG        ,MWDA        ,MWDB        , ARCDAT
     *MDB         ,XCGR        ,ZCGR        ,XE          ,ZE          ,XT          , ARCDAT
     *DREF        ,MCND        ,RHOB        ,QMULT       ,REMAX                     ARCDAT
     * ,FRATE   ,  ARCD(9)                                              RETAP
      DIMENSION  ARCDA(40)                                              ARCDAT
      EQUIVALENCE(SREF,ARCDA)                                           ARCDAT
      FTABB = 0.                                                        THRUST
      FTABO = 0.                                                        THRUST
      CALL READMS (9,SREF,40,1)                                         CKOUT
      ITAB=MT                                                           CKOUT
      IF (SW(17).EQ.0..OR.SQ(20,1).LE.0..OR.SE(2).EQ.1.) GO TO 6        CKOUT
      IF (MT GT.0.AND.SQ(20,1).LE 0.) CALL SPLIZ(MT,0,FTABB,DUM)        CKOUT
    6 CONTINUE                                                          CKOUT
      IORBI = SQ(1,3)                                                   CKOUT
      CALL READMS (9,SREF,40,IORBI)                                     CKOUT
      IF(MT.LE.0) GO TO 5                                               CKOUT
      CALL SPLIZ(MT,0,FTABO,DUM)                                        CKOUT
    5 CONTINUE                                                          CKOUT
      IF( FTABB.NE.0 ) SCB(129) = FTABB/TB36                            UH
      IF( FTABO .NE. 0 ) SCO(129) = FTABO/TO36                          UH
      IF(MT.GT.0)  GO TO 1                                              UG
C                                                                       THRUST
C          COMMON ENGINES USED                                          THRUST
C                                                                       THRUST
      IF(SE(3).GE.1.) GO TO 1                                           THRUST
```

```
      C                                                              THRUST
      C        CALCULATE ORBITER UNIT THRUST FROM BOOSTER THRUST     THRUST
      C                                                              THRUST
            FUNIT    = SCB(129)                                      CKOUT
            GO TO 2                                                  THRUST  2
      C                                                              THRUST
      C        CALCULATE ORBITER UNIT THRUST                         THRUST
      C                                                              THRUST
          1 FUNIT    = SCO(129)                                      CKOUT
      C                                                              THRUST
      C        CALCULATE BOOSTER AND ORBITER THRUST                  THRUST
      C                                                              THRUST
          2 FVACO   = FUNIT* T036 *SE(8)                             UH
            FSLO   = FVACO - SCO(218) *T036 *2116.217                UH
            FVACB   = SCB(129) * TB36 * SE(9)                        UH
            FSLB   = FVACB - SCB(218) * TB36 *2116.217               UH
      C                                                              THRUST
      C        CHECK PARALLEL BURN FLAG                              THRUST
      C                                                              THRUST
            IF(ITAB.GT.0.AND.SQ(20,1).LE.0.) GO TO 3                 CKOUT   3
            IF(SE(2).GE.2.) GO TO 3                                  THRUST  3
      C                                                              THRUST
      C        ADD THRUST LEVELS FOR LIFT-OFF THRUST                 THRUST
      C                                                              THRUST
            FVACLO   = FVACO + FVACB                                 THRUST
            FSLLO    = FSLO + FSLB                                   THRUST
            GO TO 4                                                  THRUST  4
      C                                                              THRUST
      C        DETERMINE LIFTOFF THRUST                              THRUST
      C                                                              THRUST
          3 FVACLO   = FVACB                                         THRUST
            FSLLO    = FSLB                                          THRUST
      C                                                              THRUST
      C        CHECK SOLID MOTOR FLAG                                THRUST
      C                                                              THRUST
          4 IF(SQ(20,1).LE.0.) RETURN                                CKOUT
      C                                                              THRUST
      C        DETERMINE SOLID MOTOR THRUST AND NEW LIFT-OFF THRUST  THRUST
      C                                                              THRUST
            FVACS    =   SQ(20,2) * SQ(20,1)                         THRUST
            FSLS     = FVACS - SQ(21,1)*SQ(20,1)*2116.217            THRUST
            FVACLO   =   FVACLO + FVACS                              THRUST
            FSLLO    =   FSLLO + FSLS                                THRUST
            F1     = FVACO                                           CKOUT
            T2     = SQ(21,2)                                        CKOUT
            F2     = F1 - T2* SQ(20,3)                               CKOUT
            CALL THRUP(ITAB,F1,T2,F2)                                CKOUT
      C                                                              CKOUT
      C        ESTABLISH QUANITIES FOR TWO ENGINE SIMULATION         CKOUT
      C                                                              CKOUT
            IF(SQ(30,1).NE.0.) GO TO 10                              CKOUT   10
            SQ(31,1) = FVACB                                         CKOUT
            SQ(31,2) = FVACS                                         CKOUT
            SQ(31,3) = SQ(20,4)/SE(1)                                CKOUT
            SQ(31,4) = SE(1)                                         CKOUT
            RETURN                                                   JULY28
      C                                                              CKOUT
      C        EXTERNAL TANK OPTION USED                             CKOUT
      C                                                              CKOUT
         10 CONTINUE                                                 CKOUT
            SQ(31,1) = FVACO                                         CKOUT
            SQ(31,2) = FVACS                                         CKOUT
            SQ(31,3) = SQ(20,4)/SE(7)                                CKOUT
            SQ(31,4) = SE(7)                                         CKOUT
      C                                                              THRUST
      C        RETURN TO CALLING PROGRAM                             THRUST
      C                                                              THRUST
        999 RETURN                                                   THRUST
            END                                                      THRUST
```

1575

SUBRØUTINE
VEHDF

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
A		M	Booster sub-sonic l/d	/DATA2X/(	1)	SSSP VEHDF VEHDF	O M I	A A ALD
ALD		I	Booster sub-sonic l/d	/DATA2X/(	1)	SSSP VEHDF VEHDF	O M I	A A ALD
ALD1		I	Booster flyback cruise velocity	/DATA2X/(	52)	VEHDF	I	ALD1
ALD2		I	Flag for crossfeed of propellants from booster tanks to orbiter engines at lift-off if fire =1	/DATA2X/(	53)	VEHDF	I	ALD2
ALD3		I	Data/ print flag	/DATA2X/(	54)	VEHDF	I	ALD3
AS		I	Number of allowable synthesis iterat -ions (max = 6)	/DATA2X/(	32)	VEHDF	I	AS
BECO		I	Booster cut-off arc	/SIZING/(	314)	SIZE VEHDF	I I	BECO BECO
BS		I	Booster thrust multiplier for ascent	/DATA2X/(	33)	VEHDF	I	BS
BSTG		I	Booster staging arc	/SIZING/(	315)	SIZE VEHDF	I I	BSTG BSTG
CB		I	Orbiter mass ratio tolerance input	/DATA2X/(	44)	VEHDF	I	CB
CLVG		I	Specific fuel consumption of booster air breathers	/DATA2X/(	29)	VEHDF	I	CLVG
DATA2		I	Namelist input block	/VEHDF /(*	)	VEHDF	I	DATA2
DRNG		I	Output flag for sizing data	/DATA2X/(	30)	VEHDF	I	DRNG
FBPAR		I	Estimate of slope for booster cruise adjustment if moreq 70 or mporeq 70	/DATA2X/(	2)	VEHDF	I	FBPAR
HIPSMX		I	Maximum number of iterations	/SIZING/(	292)	SIZE SSSP VEHDF VEHDF	M I I O	IPSMAX IPSMAX HIPSMX IPSMAX
ID		I	A four word array containing the basic deck, reference run, case and part case numbers in that order.	/GLOBAL/(	21)	BLICO FRENCH GEINP PADS1 PRINT SDINP TOPM VEHDF	I I I O I I I I	ID ID ID ID ID ID ID ID
IDVELO		I	Total ideal velocity estimate to parking orbit insertion	/DATA2X/(	3)	VEHDF	I	IDVELO
IPSMAX		O	Maximum number of iterations	/SIZING/(	292)	SIZE SSSP VEHDF VEHDF	M I I O	IPSMAX IPSMAX HIPSMX IPSMAX
NXFOB		I	Booster effective isp estimating parameter	/DATA2X/(	26)	VEHDF	I	NXFOB
ORBI		I	Orbiter ignition arc	/SIZING/(	316)	REU3 SIZE VEHDF	I I I	ORBI ORBI ORBI
PERISP		I	Orbiter sea level specific impulse	/DATA2X/(	8)	VEHDF	I	PERISP
RT		I	Ratio of booster to orbiter engine thrust (vac)	/DATA2X/(	46)	VEHDF	I	RT
R1		I	Desired value of l.o. T/w	/DATA2X/(	47)	VEHDF	I	R1
R3		I	Maximum number of iterations to obtain twlo	/DATA2X/(	48)	VEHDF	I	R3
SCB		I	Working name for input c-array booster scaling coefficients	/ORBINV/(	144)	FLYBKP STORE SUMOUT TAMPER THRUST VEHDF WTVOL	M M I I M I M	SCB SCB SCB SCB SCB SCB SCB

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
SE		M	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SFC1		I	Output print flag	/DATA2X/(	49)	VEHDF	I	SFC1
SFC2		I	Ascent burn sequence flag 1= simultaneous stage burn 2= sequential stage burn	/DATA2X/(	50)	VEHDF	I	SFC2
SOCO		I	Solid engine cut-off arc	/SIZING/(	326)	VEHDF	I	SOCO
SOLID		I	Number of copies of summary sheet to be output	/DATA2X/(	31)	VEHDF	I	SOLID
SOSP		I	Solid engine drop arc	/SIZING/(	327)	VEHDF	I	SOSP
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		M	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TB27		M	Stored booster value of isp(I)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		I	Stored booster value of mr(I)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TFCTRB		I	Booster vac. Isp	/DATA2X/(	15)	VEHDF	I	TFCTRB
TO27		M	Stored orbiter value of isp(I)	/ORBINX/(	41)	SIZEMR	I	TO27
						SSSP	I	TO27
						STORE	M	TO27
						SUMOUT	I	TO27
						VEHDF	M	TO27
						WTVOL	I	TO27

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
T034		O	Stored orbiter value of mr(I)	/ORBINX/(	53)	ITER8	M	T034
						SSSP	O	T034
						STORE	M	T034
						TAMPER	I	T034
						VEHDF	O	T034
						WTVOL	M	T034
TRAFLG		O	Traffic control flag 0. Sizing loop not converged 1. Sizing loop converged 2. Error in sizing loop next case	/SIZING/(	284)	FRENCH	O	TRAFLG
						ITER8	O	TRAFLG
						PADSI	I	TRAFLG
						SIZE	M	TRAFLG
						SSSP	O	TRAFLG
						VEHDF	O	TRAFLG
TRATIO		I	Orbiter vac. Isp	/DATA2X/(	19)	VEHDF	I	TRATIO
TSBD		I	Orbiter thrust multiplier for ascent	/DATA2X/(	37)	VEHDF	I	TSBD
VEHDF		E	Subroutine to call and initialize synthesis data	/VEHDF /($	)	DATAIN	S	VEHDF
						VEHDF	E	VEHDF
WFLYX		I	Tolerance on twlo iteration	/DATA2X/(	45)	VEHDF	I	WFLYX
UN06.		O	File of all output data	/.UN06./($	)	BLICO	O	.UN06.
						BNDRYC	O	.UN06.
						CRASH	O	.UN06.
						FRENCH	O	.UN06.
						FXDAT	O	.UN06.
						GEINP	O	.UN06.
						HUNT	O	.UN06.
						INEDIT	O	.UN06.
						ITER8	O	.UN06.
						MODELA	O	.UN06.
						MOMJ	O	.UN06.
						MPSI	O	.UN06.
						OUT	O	.UN06.
						PAYO2	O	.UN06.
						PRINT	O	.UN06.
						PRINTV	O	.UN06.
						PRINTW	O	.UN06.
						PRITEO	O	.UN06.
						PRITVA	O	.UN06.
						PROPIN	O	.UN06.
						PROTHR	O	.UN06.
						PRWTSM	O	.UN06.
						RANGE	O	.UN06.
						S	O	.UN06.
						SDINP	O	.UN06.
						SIZE	O	.UN06.
						SIZIN	O	.UN06.
						SIZOUT	O	.UN06.
						SOLVE	O	.UN06.
						SPLICO	O	.UN06.
						SPLIZ	O	.UN06.
						SPLYNE	O	.UN06.
						SSSP	O	.UN06.
						STAU	O	.UN06.
						STPIT	O	.UN06.
						SUMOUT	O	.UN06.
						TABIN	O	.UN06.
						TEST	O	.UN06.
						VEHDF	O	.UN06.
						WTSCH	O	.UN06.
						WTVOL	O	.UN06.

```
      SUBROUTINE VEHDF                                                  VEHDF
C                                                                       VEHDF
C          SUBROUTINE READS SYNTHESIS DATA AND SETS WORKING VARIABLES   VEHDF
C               * ADJUSTS ORBITER MASS RATIO                            VEHDF
C               * SOLID MOTOR OPTION WTS AND THRUST                     VEHDF
C                                                                       VEHDF
C          ARRAY B(57) IS USED TO OBTAIN THE SYNTHESIS                  VEHDF
C          DATA FROM THE PADS PROGRAM                                   VEHDF
C                                                                       VEHDF
      DIMENSION A(57)                                                   CKOUT
      LOGICAL NEGO                                                      CKOUT
      EQUIVALENCE(ALD,A(1))                                             CKOUT
     *,(IPSMAX,HIPSMX)                                                  UH
      REAL NXFOB                                                        DATA2X
      REAL     IVACO,    IVACB,    ISLO,     ISLB,     IDVELO           DATA2X
      COMMON/DATA2X/                                                    DATA2X
     *ALD, FBPAR, IDVELO, ISLB, ISLO, IVACB, IVACO, PERISP, QMXX,       DATA2X
     1 QMXS,SFC,SLVOUT,COPIES,SYNIT,TFCTRB,TFCTRO,TOLMU,TOLTW,          DATA2X
     2 TRATIO,TWLO,TWLOI,WTOUT,FIRE,BOOTW,VCRUSE                        DATA2X
     3 ,NXFOB,PRNTX,FSEC,CLVG,DRNG                                      DATA2X
     4 ,SOLID,AS,BS,SISP,SINERT,SAE,TSBO,FLYBCK                         DATA2X
     5 ,WPOREQ,WOREQ,GWREQ                                              DATA2X
     6 ,FBFUEL,CA,CB,WFLYX,RT,R1,R3,SFC1,SFC2,SFC3,ALD1,ALD2,ALD3,VFLY1, DATA2X
     7 VFLY2,VFLY3                                                      DATA2X
     *,TWOX(2),PNDX(10),BLOW(10),BUPP(10),STEP(11),PAYX                 POW
      EQUIVALENCE (RVAR,TWOX(2))                                        POW
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C           PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,       VV(3),      QP(14),   EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),   SQ(37,5),  SE(11),    TLAT,     TLNG,                   SIZING
C          PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,      WLOO,      DWEB,      DWEO,      TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,      BK2,       BK3,       BK4,       ISIZE,     TRAFLG,   TWRATO, SIZING
     *OK1,      OK2,       OK3,       OK4,       PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,    TVACO,     NO,        WFO,       IDVEL,     ISPO,     ISPB,   SIZING
     *XPL,      TVACB,     NNB,       WEO,       WEB,       WO,       WLO,    SIZING
     *DVO,       DVB,       MUB,       MUO,       VSTG,      WPD            SIZING
     *,JTYP      BECO       BSTG       ORBI       ITNBW   , ITNOW  ,        SIZING
     * SVDPSQ   , SVDCON    ,IHUNT     ,IOPSTG   ,ISZD(19)                UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                    CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,    ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,IO22,IO23,IO24,IO25,IO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO,SCO,OWSAVE                         CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,     UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,     UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                    CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,    ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                               UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,     UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,      UH
     8TB83,TB84                                                         UH
      COMMON/GLOBAL/                                                    GLOBAL
     *GR          ,ER         ,OMGZ       ,XLAMRF     ,YMURF     ,LUM   GLOBAL
     *,JJOP(10)  ,IFATAL      ,NARC        ,NBRAN     ,NFARC     ,ID(4) GLOBAL
     *,KTAB(20),ITAB(20)    ,SIG      ,MAXTAB                           GLOBAL
     *,GM          ,PSIRF,IPFLG1,      IPFLG2,IPFLG3,IPFLG4,INEQFL(20)  GLOBAL
     *,ITPSO  ,KSOL      ,KGLOBL(8)                                     RETAP
      COMMON/PRESET/ PRESET(57)                                         CKOUT
      COMMON/PO/MPRNT,MPNCH                                             VEHDF
      NAMELIST/DATA2/ALD,FBPAR,IDVELO,ISLB,ISLO,IVACB,IVACO,PERISP,QMXX, CKOUT
     1 QMXS,SFC,SLVOUT,COPIES,SYNIT,TFCTRB,TFCTRO,TOLMU,TOLTW,          VEHDF
     2 TRATIO,TWLO,TWLOI,WTOUT,FIRE,BOOTW,VCRUSE                        VEHDF
     3 ,NXFOB,PRNTX,FSEC, CLVG, DRNG                                    VEHDF
     4 ,SOLID,AS,BS,SISP,SINERT,SAE,TSBO  ,FLYBCK                       VEHDF
```

```
     5 ,WPOREQ,WOREQ,GWREQ                                              VEHDF
     6 ,FBFUEL,CA,CB,WFLYX,RT,R1,R3,SFC1,SFC2,SFC3,ALD1,ALD2,ALD3,VFLY1, VEHDF
     7 VFLY2,VFLY3                                                      VEHDF
      EQUIVALENCE (SOCO,ISZD(4)),(SOSP,ISZD(5))                         JULY28
C                                                                       CKOUT
C          PRESET TEST EQUATION                                         CKOUT
C                                                                       CKOUT
      NEGO(X)  =  ABS(X).EQ.0..AND.SIGN(1.,X).LT.0.                     CKOUT
C                                                                       CKOUT
      IPC = 33                                                          CKOUT
      CALL TABIN(DUM,1,A,101,RUMMY,1,10,IPC,0,IEOD)                     POW
      IF(IEOD.NE.0) GO TO 1                                             CKOUT
      DO 3 I=1,57                                                       CKOUT
      IF(NEGO(A(I)))  A(I)= PRESET(I)                                   CKOUT
    3 CONTINUE                                                          CKOUT
C ****  READ SYNTHESIS DATA  ****                                       VEHDF
      IF(MPRNT.EQ.1) WRITE(6,DATA2)                                     VEHDF
C                                                                       VEHDF
C          ESTABLISH SYNTHESIS WORKING ARRAYS                           VEHDF
C                                                                       VEHDF
      IPSMAX=NIPSMX                                                     UH
      SE(1) = IVACB                                                     VEHDF
      SE(2) = FIRE + .1                                                 VEHDF
      SE(3) = BOOTW                                                     VEHDF
      SE(4) = QMXS                                                      VEHDF
      SE(5) = FBPAR                                                     VEHDF
      IF( FLYBCK .EQ.3.) SE(5)=0.                                       VEHDF
      SE(6) = QMXX                                                      VEHDF
      SE(7) = IVACO                                                     VEHDF
      SE(8) = TFCTRO                                                    VEHDF
      SE(9) = TFCTRB                                                    VEHDF
      SE(10)= NXFOB                                                     VEHDF
      SW(2) = .5                                                        VEHDF
      SW(4) = SYNIT                                                     VEHDF
      SW(5) = TOLMU                                                     VEHDF
      SW(6) = TRATIO                                                    VEHDF
      SW(7) = PERISP                                                    VEHDF
      SW(9) = CLVG                                                      VEHDF
      SW(11) = ALD                                                      VEHDF
      SW(12) = SFC                                                      VEHDF
      SW(13) = SLVOUT                                                   VEHDF
      SW(15)=0.                                                         VEHDF
      SW(14) = VCRUSE                                                   VEHDF
      SW(16) = WTOUT                                                    VEHDF
      SW(17) = TWLO                                                     VEHDF
      SW(18) = TOLTW                                                    VEHDF
      SW(19) = TWLOI                                                    VEHDF
      SV(2) = IDVELO                                                    VEHDF
      SV(3) = SV(2)                                                     CKOUT
      SV(15) = COPIES                                                   VEHDF
      SQ(1,1) = BECO                                                    UH
      SQ(1,2) = BSTG                                                    UH
      SQ(1,3) = ORBI                                                    UH
      SQ(1,4)=SOCO                                                      JULY28
      SQ(1,5)=SOSP                                                      JULY28
      SQ(10,1) = WOREQ                                                  VEHDF
      SQ(10,3) = DRNG                                                   VEHDF
      SQ(16,1) = GWREQ                                                  VEHDF
      SQ(19,1)=0.                                                       VEHDF
      SQ(19,2)=0.                                                       VEHDF
      IF(FBFUEL.EQ.1.) SQ(19,3) = SW(11)/SW(12) * SW(14)/1.689          VEHDF
      SQ(13,1) = WPOREQ                                                 VEHDF
      SQ(19,5) = FLYBCK                                                 VEHDF
      SQ(32,1) = FBFUEL                                                 VEHDF
      SQ(32,2) = CA                                                     VEHDF
      SQ(32,3) = CB                                                     VEHDF
      SQ(32,4) = WFLYX                                                  VEHDF
      SQ(32,5) = RT                                                     VEHDF
      SQ(33,1) = R1                                                     VEHDF
      SQ(33,2) = R3                                                     VEHDF
      SQ(33,3) = SFC1                                                   VEHDF
      SQ(33,4) = SFC2                                                   VEHDF
      SQ(33,5) = SFC3                                                   VEHDF
      SQ(34,1) = ALD1                                                   VEHDF
      SQ(34,2) = ALD2                                                   VEHDF
```

```
      SQ(34,3) = ALO3                                                  VEHDF
      SQ(34,4) = VFLY1                                                 VEHDF
      SQ(34,5) = VFLY2                                                 VEHDF
      SQ(35,1) = VFLY3                                                 VEHDF
      IF(FBFUEL.EQ.2. .OR. FBFUEL.EQ.3.) SQ(19,3)= SQ(34,5)/1.689 *    VEHDF
     1 SQ(34,2) / SQ(33,4)                                             VEHDF
      TO27(3) = SE(7)                                                  VEHDF
      TB27(2) = SE(1)*(1.-2116.217*SCB(218)/SCB(129))                  VEHDF
      TB27(3) = (1.-SW(7))* TB27(2) + SW(7)* SE(1)                     VEHDF
C                                                                      VEHDF
C      ADJUST   MASS  RATIO FOR ORBITER                                VEHDF
  400 TO34(3) = EXP( (SV(2) - TB27(3) * 32.174099 * ALOG(TB34(3)  )  ) / VEHDF
     1 (TO27(3) * 32.174099)  )                                        VEHDF
C                                                                      VEHDF
C     FIXED SOLID STRAPON SYNTHESIS OPTION   ************************  VEHDF
      SQ(20,1) = SOLID                                                 VEHDF
      SQ(20,2) = AS                                                    VEHDF
      SQ(20,3) = BS                                                    VEHDF
      SQ(20,4) = SISP                                                  VEHDF
      SQ(20,5)=SINERT                                                  VEHDF
      SQ(21,1) = SAE                                                   VEHDF
      SQ(21,2) = TSBO                                                  VEHDF
      IF(SOLID.LE.0.) RETURN                                           VEHDF
      SQ(21,3) = (SOLID/SISP)*TSBO*(AS+.5*BS*TSBO)                     VEHDF
      SQ(21,4) =(AS - 14.69 * SAE  ) * SOLID                           VEHDF
      SQ(21,5)=SINERT*SOLID+SQ(21,3)                                   VEHDF
      SQ(22,1) = SQ(21,3) / SOLID                                      VEHDF
      SQ(22,2) = SQ(21,4) / SOLID                                      VEHDF
      SQ(22,3) = SOLID * SQ(20,5)                                      VEHDF
      SQ(22,4) = SOLID * SQ(20,2)                                      VEHDF
      SQ(22,5) = SOLID* SQ(21,1)                                       VEHDF
C     FIXED SOLID STRAPON SYNTHESIS OPTION   ************************  VEHDF
C                                                                      VEHDF
      RETURN                                                           VEHDF
    2 FORMAT(35H ***ERROR***  DATAZ IS MISSING      )                  UH
    1 WRITE(6,2)                                                       UH
      TRAFLG=2.                                                        UH
       CALL SIZERR                                                     UH
      END                                                              VEHDF
```

20 OCT 72 6.01-46

SUBRØUTINE
WTSCH

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
ABFSYS		M	Airbreathing fuel system weight used by set0 to set common to zero	/WTCALC/(	1)	SETO WTSCH	O M	C ABFSYS
ANENGS		I	Number of air breathing engines used by set0 to set common to zero	/CINPUT/(	1)	FRENCH FRENCH SETO STORE WTSCH	I M O M I	A ANENGS A ANENGS ANENGS
ANTANK		I	Number of air breathing fuel tanks	/CINPUT/(	2)	STORE WTSCH	M I	ANTANK ANTANK
ASRATO		I	Wing aspect ratio	/CINPUT/(	3)	STORE WTSCH	M I	ASRATO ASRATO
ASWEEP		I	Wing leading edge sweep angle	/CINPUT/(	4)	STORE WTSCH	M I	ASWEEP ASWEEP
BBODY		M	Body width used by set0 to set common to zero	/VOLCAL/(	1)	PROTHR SETO WTSCH	I O M	BBODY D BBODY
C		I	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW PRITEO PRITVA STORE WTSCH WTVOL	I I I M I O	C C C C C C
CBBODY		I	Body width coeff.	/CINPUT/(	305)	PRITVA STORE WTSCH	I M I	CBBODY CBBODY CBBODY
CFUEL		M	Mixture ratio	/CINPUT/(	306)	PRWTSM STORE WTSCH	M M M	CFUEL CFUEL CFUEL
CHBODY		I	Body height or coeff	/CINPUT/(	312)	PRITVA STORE WTSCH	I M I	CHBODY CHBODY CHBODY
CLBODY		I	Body length or coeff	/CINPUT/(	313)	PRITVA STORE WTSCH	I M I	CLBODY CLBODY CLBODY
CRAP		W	Temporary storage	/WTSCH /(*	)	WTSCH	W	CRAP
CROOT		M	Ing root chord	/VOLCAL/(	2)	PROTHR WTSCH	I M	CROOT CROOT
CSBODY		I	Total body wetted area or coeff	/CINPUT/(	314)	PRITVA STORE WTSCH	I M I	CSBODY CSBODY CSBODY
CSFAIR		I	Fairing planform area or coeff	/CINPUT/(	315)	PRITVA STORE WTSCH	I M I	CSFAIR CSFAIR CSFAIR
CSFUTK		I	Fuel tank surface area or coeff	/CINPUT/(	316)	PRITVA STORE WTSCH	I M I	CSFUTK CSFUTK CSFUTK
CSHORZ		I	Horizontal stabalizer planform area	/CINPUT/(	317)	PRITVA STORE WTSCH	I M I	CSHORZ CSHORZ CSHORZ
CSOXTK		I	Oxidizer tank surface area coeff	/CINPUT/(	318)	PRITVA STORE WTSCH	I M I	CSOXTK CSOXTK CSOXTK
CSPAN		M	Structural span along 0.5 chord	/VOLCAL/(	3)	PROTHR WTSCH	I M	CSPAN CSPAN
CSPLAN		I	Body planform area or coeff	/CINPUT/(	319)	PRITVA STORE WTSCH	I M I	CSPLAN CSPLAN CSPLAN
CSVERT		I	Vertical fin planform area or coeff	/CINPUT/(	320)	PRITVA STORE WTSCH	I M I	CSVERT CSVERT CSVERT

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
CTHRST		I	Vac. Thrust-to-weight ratio	/CINPUT/(	322)	PRITVA	I	CTHRST
						STORE	M	CTHRST
						WTSCH	I	CTHRST
						WTVOL	M	CTHRST
CTHST2		I	Secondary propulsion t/w	/CINPUT/(	323)	PRITVA	I	CTHST2
						STORE	M	CTHST2
						WTSCH	I	CTHST2
CTIP		O	Wing tip chord	/VOLCAL/(	4)	WTSCH	O	CTIP
FVACB		M	Booster vacuum thrust (lb)	/TRUST /(	5)	THRUST	M	FVACB
						WTSCH	M	FVACB
FXWOVS		I	Fixed wing loading	/CINPUT/(	329)	STORE	M	FXWOVS
						WTSCH	I	FXWOVS
GAL		M	Total gallons of fuel	/VOLCAL/(	5)	WTSCH	M	GAL
GSPAN		M	Geometric wing span	/VOLCAL/(	6)	WTSCH	M	GSPAN
HBODY		O	Body height	/VOLCAL/(	7)	PROTHR	I	HBODY
						WTSCH	O	HBODY
ISP		I	Specific impulse	/CINPUT/(	330)	PRWTSM	I	ISP
						STORE	M	ISP
						WTSCH	I	ISP
						WTVOL	O	ISP
ITPS		M	Thermo protection flag	/CINPUT/(	336)	FRENCH	O	ITPS
						STORE	M	ITPS
						WTSCH	M	ITPS
JUMP		I	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH	O	JUMP
						PRINTW	I	JUMP
						PRITVA	I	JUMP
						PRWTSM	M	JUMP
						WTSCH	I	JUMP
						WTVOL	M	JUMP
LBODY		M	Body length	/VOLCAL/(	8)	PROTHR	I	LBODY
						TAMPER	I	LBODY
						WTSCH	M	LBODY
LF		I	Ultimate load factor 1. Thrust buildup 2. Not used 3. Main impulse mass ratio 4. Main impulse reserve 5. Secondary impulse mass ratio 6. Not used	/CINPUT/(	368)	STORE	M	LF
						WTSCH	I	LF
MR		M	Mass ratio	/CINPUT/(	369)	PRWTSM	I	MR
						SOLVE	I	MR
						STORE	M	MR
						WTSCH	M	MR
NCREW		I	Number of crew members	/CINPUT/(	375)	PRITVA	I	NCREW
						STORE	M	NCREW
						WTSCH	I	NCREW
NENGS		I	Total number engines per stage	/CINPUT/(	376)	PRITVA	I	NENGS
						STORE	M	NENGS
						WTSCH	I	NENGS
NLISTO		I	Namelist output flag	/CINPUT/(	377)	STORE	M	NLISTO
						WTSCH	I	NLISTO
NPASS		I	Number of passengers	/CINPUT/(	378)	STORE	M	NPASS
						WTSCH	I	NPASS
NWL		M	Wing loading flag	/CINPUT/(	379)	FRENCH	O	NWL
						STORE	M	NWL
						WTSCH	M	NWL
PCHAM		I	Main rocket engine chamber pressure	/CINPUT/(	380)	STORE	M	PCHAM
						WTSCH	I	PCHAM
Q		M	Maximum dynamic pressure	/CINPUT/(	381)	WTSCH	M	Q
RHOFU		I	Fuel density	/CINPUT/(	382)	PRITVA	I	RHOFU
						STORE	M	RHOFU
						WTSCH	I	RHOFU
RHOFU2		I	Secondary fuel desnity	/CINPUT/(	383)	PRITVA	I	RHOFU2
						STORE	M	RHOFU2
						WTSCH	I	RHOFU2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
RHOX		I	Oxidizer density	/CINPUT/(	384)	PRITVA STORE WTSCH	I M I	RHOX RHOX RHOX
RHOX2		I	Secondary oxidizer density	/CINPUT/(	385)	PRITVA STORE WTSCH	I M I	RHOX2 RHOX2 RHOX2
RTOD		M	Deg to rad conversion	/VOLCAL/(	9)	WTSCH	M	RTOD
SBODY		M	Total body wetted area	/CINPUT/(	386)	PROTHR TAMPER WTSCH	I I M	SBODY SBODY SBODY
SFAIR		M	Total fairing or shroud surface area	/VOLCAL/(	10)	WTSCH	M	SFAIR
SFUTK		M	Total fuel tank wetted area	/VOLCAL/(	11)	PROTHR WTSCH	I M	SFUTK SFUTK
SHORZ		M	Horizontal stabilizer planform area	/VOLCAL/(	12)	PROTHR WTSCH	I M	SHORZ SHORZ
SOXTK		M	Total oxidizer tank wetted area	/VOLCAL/(	13)	PROTHR WTSCH	I M	SOXTK SOXTK
SPLAN		O	Body planform area	/VOLCAL/(	14)	PROTHR TAMPER WTSCH WTVOL	I I O I	SPLAN SPLAN SPLAN SPLAN
SQRT		F	Square root function	/SQRT /($	)	ANLATM CRASH DCTOE DER3A ENVPRM HUNT MODELA MODELB OPWELL OUT PAT63 PAY02 PDBC PDY3A STORE SYMVRT WTSCH	F F F F F F F F F F F F F F F F F	SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT SQRT
STPS		M	Total thermal protection system surface area	/VOLCAL/(	15)	WTSCH	M	STPS
SVERT		M	Vertical fin planform area	/VOLCAL/(	16)	PROTHR WTSCH	I M	SVERT SVERT
SWING		M	Gross wing area	/VOLCAL/(	17)	PROTHR STORE WTSCH	I M M	SWING SWING SWING
SXPOS		O	Exposed wing area	/VOLCAL/(	18)	PROTHR WTSCH	I O	SXPOS SXPOS
TAN		F	Tangent function	/TAN /($	)	WTSCH	F	TAN
TDEL		M	Gimbal system delivered torque	/VOLCAL/(	19)	WTSCH	M	TDEL
TOVERC		I	Wing thickness over choord ratio	/CINPUT/(	388)	PROTHR STORE WTSCH	I M I	TOVERC TOVERC TOVERC
TPRATO		I	Wing taper ratio	/CINPUT/(	389)	STORE WTSCH	M I	TPRATO TPRATO
TROOT		M	Theoritical root thickness	/VOLCAL/(	20)	WTSCH	M	TROOT
TTOT		M	Total stage vac. Thrust	/VOLCAL/(	21)	PRITVA STORE WTSCH WTVOL	I I M I	TTOT TTOT TTOT TTOT
TTOTAL		O	Total stage vac. Thrust / 1,000,000	/VOLCAL/(	23)	WTSCH	O	TTOTAL
TTOT2		M	Total stage vac. Secondary thrust	/VOLCAL/(	22)	PRITVA WTSCH	I M	TTOT2 TTOT2

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
VBODY		M	Total body volume	/CINPUT/(	391)	PRINTV	M	VBODY
						SOLVE	M	VBODY
						STORE	M	VBODY
						TAMPER	I	VBODY
						WTSCH	M	VBODY
						WTVOL	I	VBODY
VBODYA		M	Total body volume less structure	/VOLCAL/(	24)	WTSCH	M	VBODYA
VBODYX		W	Body volume	/WTSCH /(*	)	WTSCH	W	VBODYX
VBODY1		M	Vbody to - 1/3 power	/VOLCAL/(	25)	WTSCH	M	VBODY1
						WTVOL	M	VBODY1
VBODY2		M	Vbody to - 2/3 power	/VOLCAL/(	26)	WTSCH	M	VBODY2
						WTVOL	M	VBODY2
VCARGO		M	Volume of cargo bay	/VOLCAL/(	27)	PRINTV	I	VCARGO
						TAMPER	I	VCARGO
						WTSCH	M	VCARGO
VCREW		M	Volume of crew compartment	/VOLCAL/(	28)	PRINTV	I	VCREW
						WTSCH	M	VCREW
VFUTK		M	Total volume of fuel tank	/VOLCAL/(	29)	PRINTV	I	VFUTK
						STORE	M	VFUTK
						TAMPER	I	VFUTK
						WTSCH	M	VFUTK
VFUTK2		M	Total volume of secondary fuel tank	/VOLCAL/(	30)	PRINTV	I	VFUTK2
						STORE	M	VFUTK2
						TAMPER	I	VFUTK2
						WTSCH	M	VFUTK2
VINSTK		M	Total tank insulation volume	/VOLCAL/(	31)	PRINTV	I	VINSTK
						WTSCH	M	VINSTK
VLGBAY		M	Volume of recovery system bay	/VOLCAL/(	32)	PRINTV	I	VLGBAY
						WTSCH	M	VLGBAY
VOXTK		M	Total volume of oxidizer tank	/VOLCAL/(	34)	PRINTV	I	VOXTK
						STORE	M	VOXTK
						TAMPER	I	VOXTK
						WTSCH	M	VOXTK
VOXTK2		M	Total volume of secondary oxidizer tank	/VOLCAL/(	35)	PRINTV	I	VOXTK2
						STORE	M	VOXTK2
						TAMPER	I	VOXTK2
						WTSCH	M	VOXTK2
VPROP		M	Volume of propulsion bay	/VOLCAL/(	36)	PRINTV	I	VPROP
						WTSCH	M	VPROP
VSTRUC		M	Volume of basic structure	/VOLCAL/(	37)	PRINTV	I	VSTRUC
						WTSCH	M	VSTRUC
WABFPS		W	Weight of jp pressurization system	/WTSCH /(*	)	WTSCH	W	WABFPS
WABFS		O	Weight of jp fuel system less tanks	/WTSCH /(*	)	WTSCH	O	WABFS
WABFTK		M	Weight of air breathing propulsion system tanks	/WTCALC/(	2)	PRINTW	I	WABFTK
						WTSCH	M	WABFTK
WABFU		M	Weight of jp fuel	/WTCALC/(	3)	PRINTW	I	WABFU
						PRWTSM	M	WABFU
						TAMPER	I	WABFU
						WTSCH	M	WABFU
WABPR		M	Weight of air breathing engines	/WTCALC/(	4)	PRINTW	I	WABPR
						WTSCH	M	WABPR
WACRES		M	Weight of attitude control fuel reserve	/WTCALC/(	5)	PRINTW	I	WACRES
						STORE	M	WACRES
						WTSCH	M	WACRES
WACS		M	Weight of attitude control system	/WTCALC/(	6)	PRINTW	I	WACS
						WTSCH	M	WACS
WACSF		W	Weight of attitude control fuel	/WTSCH /(*	)	WTSCH	W	WACSF
WACSFO		M	Weight of attitude control fuel plus oxidizer	/WTCALC/(	7)	PRINTW	I	WACSFO
						STORE	M	WACSFO
						WTSCH	M	WACSFO
WACSO		W	Weight of attitude control oxidizer	/WTSCH /(*	)	WTSCH	W	WACSO

1687

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WACSP		W	Weight of attitude control propellnt	/WTSCH /(*	)	WTSCH	W	WACSP
WACSTK		M	Weight of attitude control tankage	/WTCALC/(	8)	PRINTW	I	WACSTK
						WTSCH	M	WACSTK
WAERO		M	Weight of aerodynamic controls	/WTCALC/(	9)	PRINTW	I	WAERO
						WTSCH	M	WAERO
WAUXT		M	Weight of separation system	/WTCALC/(	10)	PRINTW	I	WAUXT
						STORE	I	WAUXT
						WTSCH	M	WAUXT
WAVIOC		O	Total weight of avionic system	/WTSCH /(*	)	WTSCH	O	WAVIOC
WBASIC		M	Total weight of basic body	/WTCALC/(	11)	PRINTW	I	WBASIC
						PROTHR	I	WBASIC
						WTSCH	M	WBASIC
WBODY		M	Total weight of body group	/WTCALC/(	12)	PRINTW	I	WBODY
						WTSCH	M	WBODY
WBPUMP		M	Weight of boost and transfer pumps	/WTCALC/(	13)	WTSCH	M	WBPUMP
WCARGO		M	Payload weight or cargo	/WTCALC/(	14)	PRINTW	I	WCARGO
						WTSCH	M	WCARGO
WCOMM		M	Communication system weight	/WTCALC/(	15)	PRINTW	I	WCOMM
						WTSCH	M	WCOMM
WCONT		M	Contingency and growth weight	/WTCALC/(	16)	PRINTW	I	WCONT
						TAMPER	I	WCONT
						WTSCH	M	WCONT
WCOVER		M	Total weight of thermal protection system cover panels	/WTCALC/(	17)	PRINTW	I	WCOVER
						WTSCH	M	WCOVER
WDECAY		M	Thrust decay propellant weight	/WTCALC/(	18)	PRINTW	I	WDECAY
						STORE	M	WDECAY
						WTSCH	M	WDECAY
WDIST1		M	Fuel system distribution weight pt1	/WTCALC/(	19)	WTSCH	M	WDIST1
WDIST2		M	Fuel system distribution weight pt2	/WTCALC/(	20)	WTSCH	M	WDIST2
WDOCK		M	Docking structure weight	/WTCALC/(	21)	PRINTW	I	WDOCK
						WTSCH	M	WDOCK
WDPLOY		M	Deployable aerodynamic device weight	/WTCALC/(	22)	PRINTW	I	WDPLOY
						WTSCH	M	WDPLOY
WDRANS		M	Fuel tank dump and drain weight	/WTCALC/(	23)	WTSCH	M	WDRANS
WDRY		M	Stage dry weight	/WTCALC/(	24)	PRINTW	I	WDRY
						TAMPER	I	WDRY
						WTSCH	M	WDRY
WEMPTY		M	Stage empty weight	/WTCALC/(	26)	WTSCH	M	WEMPTY
WENGMT		M	Engine mount weight	/WTCALC/(	27)	WTSCH	M	WENGMT
WENGS		M	Weight of rocket engines installed	/WTCALC/(	28)	PRINTW	I	WENGS
						WTSCH	M	WENGS
WENGS2		M	Weight of secondary engines	/WTCALC/(	29)	PRINTW	I	WENGS2
						WTSCH	M	WENGS2
WFAIR		M	Weight of fairings and shrouds	/WTCALC/(	30)	PRINTW	I	WFAIR
						WTSCH	M	WFAIR
WFCONT		M	Fuel system controls weight	/WTCALC/(	31)	WTSCH	M	WFCONT
WFDCAY		M	Thrust decay fuel weight	/WTCALC/(	32)	WTSCH	M	WFDCAY
WFROST		M	Frost and ice weight	/WTCALC/(	33)	PRINTW	I	WFROST
						STORE	M	WFROST
						WTSCH	M	WFROST
WFUEL		M	Fuel weight 1. Thrust build-up fuel 2. Not used 3 Main impulse fuel wt. 4. Main impulse fuel reserve 5. Secondary impulse fuel 6. Not used	/WTCALC/(	37)	PRWTSM	M	WFUEL
						WTSCH	M	WFUEL
WFUL		M	Fuel weight	/WTCALC/(	43)	PRINTW	I	WFUL
						WTSCH	M	WFUL
WFULOS		M	Vented fuel	/WTCALC/(	44)	PRINTW	I	WFULOS
						STORE	M	WFULOS
						WTSCH	M	WFULOS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WFUNCT		M	Fuel tank weight	/WTCALC/(	45)	WTSCH	M	WFUNCT
WFUOX		M	Weight of main and secondary propellant	/WTCALC/(	46)	PRINTW	I	WFUOX
						TAMPER	I	WFUOX
						WTSCH	M	WFUOX
						WTVOL	I	WFUOX
WFURES		M	Fuel reserve	/WTCALC/(	47)	PRINTW	I	WFURES
						STORE	M	WFURES
						TAMPER	I	WFURES
						WTSCH	M	WFURES
WFUSYS		M	Total fuel system weight	/WTCALC/(	48)	PRINTW	I	WFUSYS
						WTSCH	M	WFUSYS
WFUTK		M	Wt of non-structural fuel tankage	/WTCALC/(	49)	PRINTW	I	WFUTK
						WTSCH	M	WFUTK
WFUTK2		M	Wt of secondary fuel tank and system	/WTCALC/(	50)	PRINTW	I	WFUTK2
						WTSCH	M	WFUTK2
WFUTOT		M	Total weight of fuel	/WTCALC/(	51)	WTSCH	M	WFUTOT
WFUTRP		M	Trapped fuel weight	/WTCALC/(	52)	PRINTW	I	WFUTRP
						STORE	M	WFUTRP
						WTSCH	M	WFUTRP
WFU2		M	Weight of secondary fuel	/WTCALC/(	34)	PRINTW	I	WFU2
						TAMPER	I	WFU2
						WTSCH	M	WFU2
WGASPR		M	Weight of gas and pressurant	/WTCALC/(	53)	PRINTW	I	WGASPR
						STORE	M	WGASPR
						WTSCH	M	WGASPR
WGNAV		M	Guidance and navigation system wt	/WTCALC/(	54)	PRINTW	I	WGNAV
						WTSCH	M	WGNAV
WGROSS		M	Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
						PRWTSM	I	WGROSS
						SOLVE	M	WGROSS
						STORE	M	WGROSS
						TAMPER	I	WGROSS
						WTSCH	M	WGROSS
						WTVOL	I	WGROSS
WHORZ		M	Horizontal stabilizer wt.	/WTCALC/(	55)	PRINTW	I	WHORZ
						PROTHR	I	WHORZ
						WTSCH	M	WHORZ
WHYCAD		M	Hydraulic / pneumatic system wt	/WTCALC/(	56)	PRINTW	I	WHYCAD
						WTSCH	M	WHYCAD
WINFUT		M	Weight of intergral fuel tank	/WTCALC/(	57)	PRINTW	I	WINFUT
						PROTHR	I	WINFUT
						WTSCH	M	WINFUT
WINOXT		M	Weight of integral oxidizer tank	/WTCALC/(	58)	PRINTW	I	WINOXT
						PROTHR	I	WINOXT
						WTSCH	M	WINOXT
WINST		M	Weight of instrument system	/WTCALC/(	60)	PRINTW	I	WINST
						WTSCH	M	WINST
WINSTK		M	Total weight of tank insulation	/WTCALC/(	59)	PRINTW	I	WINSTK
						WTSCH	M	WINSTK
WJET		M	Jettison weight 1. Ignition to lift-off 2. Not used 3. Jettison during ascent 4. In-orbit jettison wt. 5. Pre-entry jettison wt. 6. Fly-back jettison wt.	/WTCALC/(	62)	PRITVA	I	WJET
						PRWTSM	M	WJET
						STORE	O	WJET
						TAMPER	I	WJET
						WTSCH	M	WJET
WLANCH		M	Launch gear weight	/WTCALC/(	68)	PRINTW	I	WLANCH
						WTSCH	M	WLANCH
WLG		M	Landing gear and controls weight	/WTCALC/(	69)	PRINTW	I	WLG
						WTSCH	M	WLG
WLOSS		O	In-flight weight loss	/WTCALC/(	70)	PRINTW	I	WLOSS
						PRWTSM	I	WLOSS
						WTSCH	O	WLOSS

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
WLRD		M	Launch and recovery system weight	/WTCALC/(	71)	PRINTW	I	WLRD
						WTSCH	M	WLRD
WNACEL		M	Pylons, nacel,and pod weights	/WTCALC/(	72)	PRINTW	I	WNACEL
						WTSCH	M	WNACEL
WODCAY		M	Oxidizer thrust decay weight	/WTCALC/(	73)	WTSCH	M	WODCAY
WOIL		M	Service item losses	/WTCALC/(	74)	PRINTW	I	WOIL
						WTSCH	M	WOIL
WOILRS		M	Service item reserves	/WTCALC/(	75)	PRINTW	I	WOILRS
						WTSCH	M	WOILRS
WORSUL		M	Orientation,control, and separation system weight	/WTCALC/(	76)	PRINTW	I	WORSUL
						WTSCH	M	WORSUL
WOVERS		M	Wing loading	/WTCALC/(	77)	PROTHR	I	WOVERS
						TAMPER	I	WOVERS
						WTSCH	M	WOVERS
WOX		M	Thrust build-up oxidizer 1. Thrust build-up oxidizer 2. Not used 3. Main impulse oxidizer 4. Main impulse oxidizer reserve 5. Secondary impulse oxidizer 6. Not used	/WTCALC/(	78)	PRWTSM	M	WOX
						WTSCH	M	WOX
WOXID		M	Main impulse oxidizer weighter	/WTCALC/(	87)	PRINTW	I	WOXID
						WTSCH	M	WOXID
WOXLOS		M	Vented oxidizer	/WTCALC/(	88)	PRINTW	I	WOXLOS
						STORE	M	WOXLOS
						WTSCH	M	WOXLOS
WOXRES		M	Oxidizer reserve	/WTCALC/(	89)	PRINTW	I	WOXRES
						STORE	M	WOXRES
						TAMPER	I	WOXRES
						WTSCH	M	WOXRES
WOXSYS		M	Oxidizer system weight	/WTCALC/(	90)	PRINTW	I	WOXSYS
						WTSCH	M	WOXSYS
WOXTK		M	Non-sturctural tank wt.- oxidizer	/WTCALC/(	91)	PRINTW	I	WOXTK
						WTSCH	M	WOXTK
WOXTK2		M	Secondary system oxidizer tank wt	/WTCALC/(	92)	PRINTW	I	WOXTK2
						WTSCH	M	WOXTK2
WOXTOT		M	Total weight of oxidizer	/WTCALC/(	93)	WTSCH	M	WOXTOT
WOXTRP		M	Trapped oxidizer weight	/WTCALC/(	94)	PRINTW	I	WOXTRP
						STORE	M	WOXTRP
						WTSCH	M	WOXTRP
WOX2		M	Secondary oxidizer weight	/WTCALC/(	84)	PRINTW	I	WOX2
						TAMPER	I	WOX2
						WTSCH	M	WOX2
WP		M	Total propellant weight	/WTCALC/(	95)	WTSCH	M	WP
WPASS		M	Weight of passengers	/WTCALC/(	96)	PRINTW	I	WPASS
						WTSCH	M	WPASS
WPAYL		M	Payload weight	/WTCALC/(	97)	PRINTW	I	WPAYL
						TAMPER	I	WPAYL
						WTSCH	M	WPAYL
WPERS		M	Crew gear and life support weight	/WTCALC/(	98)	PRINTW	I	WPERS
						WTSCH	M	WPERS
WPOWCD		M	Power conditioning equipment wt.	/WTCALC/(	99)	PRINTW	I	WPOWCD
						WTSCH	M	WPOWCD
WPOWFO		M	Power system propellant wt.	/WTCALC/(	101)	PRINTW	I	WPOWFO
						STORE	M	WPOWFO
						WTSCH	M	WPOWFO
WPOWRS		M	Power system propellant reserve	/WTCALC/(	102)	PRINTW	I	WPOWRS
						STORE	M	WPOWRS
						WTSCH	M	WPOWRS
WPOWTK		M	Prime power system tank weight	/WTCALC/(	103)	PRINTW	I	WPOWTK
						WTSCH	M	WPOWTK
WPPROV		M	Personnel provisions	/WTCALC/(	104)	PRINTW	I	WPPROV
						WTSCH	M	WPPROV

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	CODE	VAR
WPREIG		M	Pre-ignition losses	/WTCALC/(	105)	PRINTW PRWTSM WTSCH	I M M	WPREIG WPREIG WPREIG
WPROP		M	Total weight- propulsion group	/WTCALC/(	106)	PRINTW WTSCH	I M	WPROP WPROP
WPRSYS		M	Pressurization system weight	/WTCALC/(	107)	PRINTW WTSCH	I M	WPRSYS WPRSYS
WREFUL		M	Fuel system refueling system weight	/WTCALC/(	108)	WTSCH	M	WREFUL
WRESID		M	Weight of residuals	/WTCALC/(	109)	PRINTW WTSCH	I M	WRESID WRESID
WRESRV		M	Propellant reservs	/WTCALC/(	110)	PRINTW WTSCH	I M	WRESRV WRESRV
WSEAL		M	Fuel tank seal weight	/WTCALC/(	111)	WTSCH	M	WSEAL
WSECST		M	Secondary body structure wt	/WTCALC/(	112)	PRINTW PROTHR WTSCH	I I M	WSECST WSECST WSECST
WSORCE		M	Prime power system weight	/WTCALC/(	113)	PRINTW WTSCH	I M	WSORCE WSORCE
WSRTRP		M	Trapped oxidizer weight	/WTCALC/(	114)	PRINTW STORE WTSCH	I M M	WSRTRP WSRTRP WSRTRP
WSTAB		M	Engine gimbal system wt	/WTCALC/(	115)	PRINTW WTSCH	I M	WSTAB WSTAB
WSURF		M	Aero surface wt	/WTCALC/(	116)	PRINTW WTSCH	I M	WSURF WSURF
WTABC		O	Net stage weight	/WTCALC/(	117)	WTSCH	O	WTABC
WTHRST		M	Thrust structure wt	/WTCALC/(	118)	PRINTW WTSCH	I M	WTHRST WTHRST
WTO		M	Take- off weight	/WTCALC/(	119)	PRINTW WTSCH	I M	WTO WTO
WTPS		M	Induced environmental protection wt	/WTCALC/(	120)	PRINTW WTSCH	I M	WTPS WTPS
WTSCH		E	Subroutine to calculate weight and volume of both stages	/WTSCH /($	)	SOLVE WTSCH WTVOL	S E S	WTSCH WTSCH WTSCH
WVERT		M	Vertical fin weight	/WTCALC/(	121)	PRINTW PROTHR WTSCH	I I M	WVERT WVERT WVERT
WWAIT		M	Summary weights 1. Ignition 2. Take-off 3. Burnout 4. Initial orbiter 5. Initial entry 6. Initial flyback 7. Landing	/WTCALC/(	122)	PRITVA PRWTSM STORE TAMPER WTSCH	I I M I M	WWAIT WWAIT WWAIT WWAIT WWAIT
WWET		M	Operating weight-empty	/WTCALC/(	132)	PRINTW WTSCH	I M	WWET WWET
WWING		M	Total structural wt. Of wing	/WTCALC/(	133)	PRINTW PROTHR WTSCH	I I M	WWING WWING WWING
WZROFU		M	Zero fuel weight of vehicle	/WTCALC/(	134)	WTSCH	M	WZROFU

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
.UN06.		0	File of all output data	/.UN06./(¢	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAYO2	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      SUBROUTINE WTSCH                                                  WTSCH
C *** WTSCH  ***                                                         WTSCH
C                                                                       WTSCH
C     WTSCH SUBROUTINE REPROGRAMED BY WALTER D. HONEYCUTT                WTSCH
C     AND B. H. OMAN ON 9/17/70 INPUT  DECKS PRIOR TO THIS DATA ARE NO  WTSCH
C     LONGER VALID                                                      WTSCH
C                                                                       WTSCH
C                                                                       WTSCH
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C           PHASE II SIZING PARAMERERS                                  SIZING
     *TZ,      VV(3),     QP(14),    EROR,     PZ(5),     VQ,      SW(20), SIZING
     *SV(28),  SQ(37,5),  SE(11),    TLAT,     TLNG,                    SIZING
C          PHASE I SIZING PARAMERERS                                    SIZING
     *WBO,     WLOO,      DWEB,      DWEO,     TOLWT,     WPB,     TWRAT2, SIZING
     *BK1,     BK2,       BK3,       BK4,      ISIZE,     TRAFLG,  TWRATO, SIZING
     *OK1,     OK2,       OK3,       OK4,      PRFLG,     IPASS,   IPSMAX, SIZING
     *AEXIT,   TVACO,     NO,        WFO,      IDVEL,     ISPO,    ISPB,   SIZING
     *XPL,     TVACB,   NNB,         WEO,      WEB,       WO,      WLO,    SIZING
     *DVO,      DVB,       MUB,       MUO,     VSTG,     WPO               SIZING
     *,JTYP,    BECO,     BSTG,      ORBI,    ITNBW , ITNOW  ,             SIZING
     * SVOPSO  , SVDCON  ,IHUNT   ,IOPSTG  ,ISZO(19)                       UN
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   ,KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                          CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                          VOLCAL
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCO  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                        CKOUT
C                                                                       WTSCH
      COMMON/JUMPY/JUMP,WBIG,WBOO                                       WTSCH
      COMMON /TRUST/                                                    WTSCH
     * FVACO,   FSLO,     FVACLO,    FVACS,                             CKOUT
     *FVACB,    FSLB,     FSLLO,     FSLS                               WTSCH
C                                                                       WTSCH
      NAMELIST/MASS1/                                                   WTSCH
     1ABFSYS  ,BBODY   ,CROOT   ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   WTSCH
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   WTSCH
     3SPLAN   ,STPS    ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   WTSCH
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   WTSCH
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   WTSCH
     6VOXTK2  ,VPROP   ,VSTRUC  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTSCH
     7WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTSCH
     8WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTSCH
     9WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTSCH
     *WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST      WTSCH
      NAMELIST/MASS2/                                                   WTSCH
```

```
      *WFU2    ,WFUEL   ,WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,      WTSCH
      *WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,      WTSCH
      *WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,      WTSCH
      *WJET    ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,      WTSCH
      *WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX     ,WOX2    ,WOXID   ,      WTSCH
      *WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,      WTSCH
      *WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCD  ,WPOWER  ,WPOWFO  ,      WTSCH
      *WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,      WTSCH
      *WRESID  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,      WTSCH
      *WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT   ,      WTSCH
      *WWET    ,WWING   ,WZROFU                                             WTSCH
C                                                                           WTSCH
C      *************  END  COMMON ****************************************  WTSCH
C                                                                           WTSCH
C                                                                           WTSCH
C      INITALIZE VARIABLES                                                  WTSCH
C                                                                           WTSCH
C                                                                           WTSCH
       WTO=WGROSS                                                           WTSCH
       Q = SE(6)                                                            CKOUT
       C13=1./3.                                                            WTSCH
       C23=2./3.                                                            WTSCH
       RTOD=57.3                                                            WTSCH
       WPREIG=C(134)                                                        WTSCH
       WWAIT(1)=WGROSS-WPREIG                                               WTSCH
C                                                                           WTSCH
C         CALCULATE PROPELLANT WEIGHTS                                      WTSCH
C                                                                           WTSCH
C      COMPUTE FUEL PERCENTAGE                                              WTSCH
C                                                                           WTSCH
       DO 10 I=1,6                                                          WTSCH
C                                                                           WTSCH
       IF (CFUEL(I).GE.1.0) CFUEL(I)=1./(CFUEL(I)+1.)                       WTSCH
C                                                                           WTSCH
C      COMPUTE MASS RATIO                                                   WTSCH
C                                                                           WTSCH
       IF(ISP(I).NE.0..AND.MR(I).GT.15.)MR(I)=EXP(MR(I)/32.174/ISP(I))      WTSCH
       IF(MR(I).EQ.0.0) MR(I)=1.                                            WTSCH
C                                                                           WTSCH
    10 CONTINUE                                                             WTSCH
C                                                                           WTSCH
C                                                                           WTSCH
C      COMPUTE JETTSION WEIGHT                                              WTSCH
C                                                                           WTSCH
       WJET(1)=0.0                                                          WTSCH
       WJET(2)=0.0                                                          WTSCH
       IF(JUMP.EQ.0)                                                        WTSCH
      1 WJET(3)=WFROST + C(127)                                             WTSCH
       WJET(4)=WSRTRP + WDECAY + WFURES + WOXRES                            WTSCH
      1  +  C(296)                                                          WTSCH
       WJET(5)=WACSFD + C(76)  + WGASPR + WFULOS + WOXLOS                   WTSCH
       WJET(6)=WPOWFO - C(127)                                              WTSCH
       WFUL=0.0                                                             WTSCH
       WOXID=0.0                                                            WTSCH
C                                                                           WTSCH
C      COMPUTE PROPELLANT WEIGHT                                            WTSCH
C                                                                           WTSCH
       DO 40 I=1,4                                                          WTSCH
C                                                                           WTSCH
       WFUOX=0.                                                             WTSCH
       IF (MR(I).NE.0.) WFUOX=WWAIT(I)*(MR(I)-1.)/MR(I)                     WTSCH
C                                                                           WTSCH
C      COMPUTE FUEL WEIGHTS                                                 WTSCH
C                                                                           WTSCH
       WFUEL(I)=WFUOX*CFUEL(I)                                              WTSCH
C                                                                           WTSCH
C      COMPUTE LOX WEIGHTS                                                  WTSCH
C                                                                           WTSCH
       WOX(I)=WFUOX-WFUEL(I)                                                WTSCH
C                                                                           WTSCH
C      THRUST BUILD UP PROPELLANTS INPUT CONSTANT                           WTSCH
C                                                                           WTSCH
       IF (I.NE.1) GO TO 30                                                 WTSCH   30
C                                                                           WTSCH
       WFUEL(I)=WFUEL(I)+C(132)                                             WTSCH
```

20 OCT 72 6.01-46

```
      WOX(I)=WOX(I)+C(133)                                              WTSCH
      WFUOX=WFUEL(I)+WOX(I)                                             WTSCH
C                                                                       WTSCH
   30 CONTINUE                                                          WTSCH
C                                                                       WTSCH
C     COMPUTE VEHICLE WEIGHTS                                           WTSCH
C                                                                       WTSCH
      IF(I.EQ.4) GO TO 40                                               WTSCH
C                                                                       WTSCH
      WWAIT(I+1)=WWAIT(I)-WFUOX-WJET(I)                                 WTSCH
C                                                                       WTSCH
C     SUM  FUEL WEIGHTS                                                 WTSCH
C                                                                       WTSCH
C                                                                       WTSCH
      WFUL=WFUL+WFUEL(I)                                                WTSCH
C                                                                       WTSCH
C     SUM  LOX WEIGHTS                                                  WTSCH
C                                                                       WTSCH
      WOXID=WOXID+WOX(I)                                                WTSCH
C                                                                       WTSCH
   40 CONTINUE                                                          WTSCH
C                                                                       WTSCH
      WWAIT(5)=WWAIT(4)-WJET(4)                                         WTSCH
C                                                                       WTSCH
C     COMPUTE TOTAL THRUST                                              WTSCH
C                                                                       WTSCH
      IF( JUMP.EQ.0) TTOT= FVACO                                        WTSCH
      IF(SE(3). EQ.1.) FVACB=0.                                         UN
      IF(JUMP.EQ.1) TTOT= FVACB + CTHRST*WWAIT (3) + C(232)             UN
C                                                                       WTSCH
C     TOTAL THRUST/1000000                                              WTSCH
C                                                                       WTSCH
      TTOTAL=TTOT/1000000.                                              WTSCH
C                                                                       WTSCH
C     COMPUTE SECONDARY THRUST                                          WTSCH
C                                                                       WTSCH
      TTOT2=WWAIT(4)*CTHST2+C(158)                                      WTSCH
C                                                                       WTSCH
C     COMPUTE ACS PROPELLANTS                                           WTSCH
C                                                                       WTSCH
      WACSFO=C(173)*WTO+C(174)*WWAIT(4)+C(175)                          WTSCH
C                                                                       WTSCH
C     COMPUTE POWER SOURCE PROPELLANTS                                  WTSCH
C                                                                       WTSCH
      WPOWFO=C(38)*WWAIT(6)+C(127)                                      WTSCH
C                                                                       WTSCH
C     COMPUTE SERVICE ITEM LOSSES                                       WTSCH
C                                                                       WTSCH
      WOIL=C(130)*TTOT+C(131)                                           WTSCH
C                                                                       WTSCH
C     COMPUTE MAIN FUEL RESERVES                                        WTSCH
C                                                                       WTSCH
      WFURES=C(115)*WFUEL(3)+C(116) + WFUEL(4)                          WTSCH
C                                                                       WTSCH
C     COMPUTE MAIN LOX RESERVES                                         WTSCH
C                                                                       WTSCH
      WOXRES=C(117)*WOX  (3)+C(118) +WOX(4)                             WTSCH
C                                                                       WTSCH
C     COMPUTE ACS PROPELLANT RESERVE                                    WTSCH
C                                                                       WTSCH
      WACRES=C(172)*WACSFO+C(73)                                        WTSCH
C                                                                       WTSCH
C     COMPUTE POWER SOURCE PROPELLANT RESERVE                           WTSCH
C                                                                       WTSCH
      WPOWRS=C(119)*WPOWFO+C(120)                                       WTSCH
C                                                                       WTSCH
C     COMPUTE SERVICE ITEM RESERVES                                     WTSCH
C                                                                       WTSCH
      WOILRS=C(121)*WOIL+C(122)                                         WTSCH
C                                                                       WTSCH
C     SUM PROPELLANT RESERVES                                           WTSCH
C                                                                       WTSCH
      WRESRV=WFURES+WOXRES+WACRES +WPOWRS+WOILRS                        WTSCH
C                                                                       WTSCH
```

```
C      COMPUTE SECONDARY FUEL WEIGHT                                    WTSCH
C                                                                       WTSCH
       WFU2(1)=0.0                                                      WTSCH
       WOX2(1)=0.0                                                      WTSCH
       IF (MR(5).EQ.0.0) GO TO 45                                       WTSCH
       WFU2(1)=WWAIT(4)*(MR(5)-1.)/MR(5)*(    CFUEL(5))                 WTSCH
       WFUEL(5)=WFU2(1)                                                 WTSCH
C                                                                       WTSCH
C      COMPUTE SECONDARY LOX WEIGHT                                     WTSCH
C                                                                       WTSCH
       WOX2(1)=WWAIT(4)*(MR(5)-1.)/MR(5)*(1.-CFUEL(5))                  WTSCH
       WOX(5)=WOX2(1)                                                   WTSCH
    45 CONTINUE                                                         WTSCH
       WPOMS=WFU2(1) + WOX2(1)                                          WTSCH
C                                                                       WTSCH
C      COMPUTE TOTAL FUEL WEIGHT LESS TRAPPED AND LOSSES                WTSCH
C                                                                       WTSCH
       WFUTOT=0.0                                                       WTSCH
       IF (C(109).NE.1.0) WFUTOT=(WFUL+WFURES+C(110))/(1.-C(109))       WTSCH
C                                                                       WTSCH
C      COMPUTE TOTAL LOX WEIGHT LESS TRAPPED AND LOSSES                 WTSCH
C                                                                       WTSCH
       WOXTOT=0.0                                                       WTSCH
       IF (C(111).NE.1.0) WOXTOT=(WOXID+WOXRES+C(112))/(1.-C(111))      WTSCH
C                                                                       WTSCH
C      SUM PROPELLANT WEIGHT LESS LOSSES AND TRAPPED                    WTSCH
C                                                                       WTSCH
       WP=WFUTOT+WOXTOT                                                 WTSCH
C                                                                       WTSCH
C      COMPUTE FUEL LOSSES                                              WTSCH
C                                                                       WTSCH
       WFULOS=C(123)*WFUTOT+C(229)*WP+C(124)                            WTSCH
      1  + C(299)*WPOMS                                                 WTSCH
C                                                                       WTSCH
C      COMPUTE LOX LOSSES                                               WTSCH
C                                                                       WTSCH
       WOXLOS=C(125)*WOXTOT+C(230)*WP+C(126)                            WTSCH
C                                                                       WTSCH
C      SUM TOTAL FUEL WEIGHT AND LOSSES                                 WTSCH
C                                                                       WTSCH
       WFUTOT=WFUTOT+WFULOS                                             WTSCH
C                                                                       WTSCH
C      SUM TOTAL LOX  WEIGHT AND LOSSES                                 WTSCH
C                                                                       WTSCH
       WOXTOT=WOXTOT+WOXLOS                                             WTSCH
C                                                                       WTSCH
C      SUM PROPELLANT WEIGHT LESS TRAPPED                               WTSCH
C                                                                       WTSCH
       WP=WFUTOT+WOXTOT                                                 WTSCH
C                                                                       WTSCH
C      COMPUTE TRAPPED FUEL WEIGHT                                      WTSCH
C                                                                       WTSCH
       WFUTRP=C(109)*WFUTOT+C(225)*WP+C(226)*TTOT   +C(110)             WTSCH
      1  + C(300)*WPOMS                                                 WTSCH
C                                                                       WTSCH
C      COMPUTE TRAPPED LOX WEIGHT                                       WTSCH
C                                                                       WTSCH
       WOXTRP=C(111)*WOXTOT+C(227)*WP+C(228)*TTOT   +C(112)             WTSCH
C                                                                       WTSCH
C      COMPUTE THRUST DECAY PROPELLANTS                                 WTSCH
C                                                                       WTSCH
       WDECAY=C(166)*TTOT+C(167)                                        WTSCH
       WFDCAY=WDECAY*CFUEL(3)                                           WTSCH
       WODCAY=WDECAY-WFDCAY                                             WTSCH
C                                                                       WTSCH
C      COMPUTE TOTAL FUEL WEIGHT                                        WTSCH
C                                                                       WTSCH
       WFUTOT=WFUTOT+WFUTRP+WFDCAY                                      WTSCH
C                                                                       WTSCH
C      COMPUTE TOTAL LOX WEIGHT                                         WTSCH
C                                                                       WTSCH
       WOXTOT=WOXTOT+WOXTRP+WODCAY                                      WTSCH
C                                                                       WTSCH
C      COMPUTE TANK PRESSURIZATION AND PURGE GASES                      WTSCH
C                                                                       WTSCH
```

```
      WGASPR=C(106)*VFUTK+C(107)*VOXTK+C(108)                          WTSCH
C                                                                      WTSCH
C     COMPUTE TRAPPED SERVICE ITEMS                                    WTSCH
C                                                                      WTSCH
      WSRTRP=C(113)*WWAIT(1)+C(114)                                    WTSCH
C                                                                      WTSCH
C     COMPUTE ICE AND FROST                                            WTSCH
C                                                                      WTSCH
      WFROST=C(78)                                                     WTSCH
C                                                                      WTSCH
C                                                                      WTSCH
C     SUM ENTRY WEIGHT                                                 WTSCH
C                                                                      WTSCH
      WWAIT(6)=WWAIT(5)-WFUEL(5)-WOX(5)-WJET(5)                        WTSCH
C                                                                      WTSCH
C     COMPUTE AIR BREATHING FUEL                                       WTSCH
C                                                                      WTSCH
      WABFU=0.0                                                        WTSCH
      WABFU=C(215)*C(214)/(1.+C(214))*WWAIT(6)                         WTSCH
C                                                                      WTSCH
C     COMPUTE AIR BREATHING FUEL TANK VOLUME                           WTSCH
C                                                                      WTSCH
      IF(RHOFU.EQ.0.0.OR.WWAIT(6).EQ.0.0)  GO TO 9999                  WTSCH  9999
      IF(ANTANK.GT.0.0)  GO TO 9999                                    WTSCH  9999
      IF(C(212).NE.0.0.OR.C(213).NE.0.0) K(28)=WABFU/RHOFU             WTSCH
 9999 CONTINUE                                                         WTSCH
C                                                                      WTSCH
C     SUM LANDING WEIGHT                                               WTSCH
C                                                                      WTSCH
      WWAIT(7)=WWAIT(6) - WABFU - WJET(6)                              WTSCH
C                                                                      WTSCH
C     SUM SUBTOTAL                                                     WTSCH
C                                                                      WTSCH
      WTO=WWAIT(1)-WFUEL(1)-WOX(1)                                     WTSCH
      WFUOX=WFUL+WOXID+WFU2(1)+WOX2(1)                                 WTSCH
      WRESID=WFUTRP+WOXTRP+WGASPR+WSRTRP                               WTSCH
      WLOSS=WFULOS+WOXLOS+WACSFO+WPOWFO+WOIL+WABFU+WFROST              WTSCH
     1 + C(296)                                                        WTSCH
C                                                                      WTSCH
C     ******                                                           WTSCH
C              GEOMETRY  SECTION                                       WTSCH
C     *****                                                            WTSCH
C                                                                      WTSCH
C                                                                      WTSCH
C     COMPUTE BODY VOLUME TO THE 1/3 POWER                             WTSCH
C                                                                      WTSCH
      IF(VBODY.LE.0.0) VBODY=1.0                                       WTSCH
      VBODY1=VBODY**C13                                                WTSCH
C                                                                      WTSCH
C     COMPUTE BODY VOLUME TO THE 2/3 POWER                             WTSCH
C                                                                      WTSCH
      VBODY2=VBODY1**2                                                 WTSCH
C                                                                      WTSCH
C     COMPUTE LOX TANK VOLUME                                          WTSCH
C                                                                      WTSCH
      VOXTK=0.0                                                        WTSCH
      IF (RHOX.NE.0.) VOXTK=(WOXTOT/RHOX)*(K(2)+1.)+K(29)              WTSCH
C                                                                      WTSCH
C     COMPUTE FUEL TANK VOLUME                                         WTSCH
C                                                                      WTSCH
      VFUTK=0.0                                                        WTSCH
      IF (RHOFU.NE.0.) VFUTK=(WFUTOT/RHOFU)*(K(1)+1.)+K(28)+K(21)      WTSCH
C                                                                      WTSCH
C     COMPUTE FUEL TANK SURFACE AREA                                   WTSCH
C                                                                      WTSCH
      SFUTK=0.0                                                        WTSCH
      IF(VFUTK.GT.0.0)                                                 WTSCH
     1SFUTK=CSFUTK*VFUTK**C23                                          WTSCH
C                                                                      WTSCH
C     COMPUTE LOX  TANK SURFACE AREA                                   WTSCH
C                                                                      WTSCH
      SOXTK=0.0                                                        WTSCH
      IF(VOXTK.GT.0.0)                                                 WTSCH
     1SOXTK=CSOXTK*VOXTK**C23                                          WTSCH
C                                                                      WTSCH
```

```
      C     COMPUTE PROPELLANT TANKS INSULATION VOLUME                          WTSCH
      C                                                                         WTSCH
            VINSTK=K(3)*SFUTK+K(4)+K(25)*SOXTK                                  WTSCH
      C                                                                         WTSCH
      C     COMPUTE SECONDARY FUEL TANK VOLUME                                  WTSCH
      C                                                                         WTSCH
            VFUTK2=K(7)                                                         WTSCH
            WFUL2=WFU2(1)                                                       WTSCH
            WOXD2=WOX2(1)                                                       WTSCH
            IF(JUMP .GT. 0.0) GO TO 38                                          WTSCH
            IF(C(270) .LE. 0.0) GO TO 37                                        WTSCH
            DVSIZ2=C(270)                                                       WTSCH
            SISP2=C(271)                                                        WTSCH
            CWINJ=C(272)                                                        WTSCH
            OFOMS=C(273)                                                        WTSCH
            R2=EXP(DVSIZ2/(32.17405*SISP2))                                     WTSCH
            WO2=CWINJ*WWAIT(4)                                                  WTSCH
            WC2=WO2/R2                                                          WTSCH
            WP2=WO2-WC2                                                         WTSCH
            WFUL2=WP2/(1.0+OFOMS)                                               WTSCH
            WOXD2=WP2-WFUL2                                                     WTSCH
         37 CONTINUE                                                            WTSCH
            IF(C(164).GT.0.0.OR.C(165).GT.0.0) GO TO 38                         WTSCH
            OFACS=C(34)                                                         WTSCH
            WACSP=WACSFO + WACRES                                               WTSCH
            WACSF=WACSP/(1.0 + OFACS)                                           WTSCH
            WACSO=WACSP - WACSF                                                 WTSCH
            WFUL2=WFUL2 + WACSF                                                 WTSCH
            WOXD2= WOXD2 + WACSO                                                WTSCH
         38 CONTINUE                                                            WTSCH
            IF(RHOFU2.NE.0.0)                                                   WTSCH
           1VFUTK2=WFUL2/RHOFU2                                                 WTSCH
      C                                                                         WTSCH
      C     COMPUTE SECONDARY OXIDIZER TANK VOLUME                              WTSCH
      C                                                                         WTSCH
            VOXTK2=K(8)                                                         WTSCH
            IF(RHOX2.NE.0.0)                                                    WTSCH
           1VOXTK2=WOXD2/RHOX2                                                  WTSCH
      C                                                                         WTSCH
      C     COMPUTE PROPULSION  SYSTEM VOLUME                                   WTSCH
      C                                                                         WTSCH
            VPROP=K(16)*TTOT+K(17)                                              WTSCH
      C                                                                         WTSCH
      C     COMPUTE CARGO VOLUME                                                WTSCH
      C                                                                         WTSCH
            VCARGO=K(9)                                                         WTSCH
      C                                                                         WTSCH
      C     COMPUTE CREW VOLUME                                                 WTSCH
      C                                                                         WTSCH
            VCREW=K(5)*NCREW+K(6)                                               WTSCH
      C                                                                         WTSCH
      C     COMPUTE LANDING GEAR BAY VOLUME                                     WTSCH
      C                                                                         WTSCH
            VLGBAY=K(12)*WLG+K(13)                                              WTSCH
      C                                                                         WTSCH
      C     COMPUTE VBODY,VOTHER,SBODY AND VSTRUC USING K(18) SCALING DATA      WTSCH
      C                                                                         WTSCH
            IF(K(18).LE.1.0 ) GO TO 46                                          WTSCH
            VBODY=K(18)*(VFUTK+VOXTK)+K(23)                                     WTSCH
            VOTHER=VBODY-VFUTK-VOXTK-VINSTK-VCREW-VCARGO-VSTRUC-VLGBAY-VPROP    WTSCH
           *-VFUTK2-VOXTK2                                                      WTSCH
            SBODY=CSBODY*VBODY**C23                                             WTSCH
            VSTRUC=K(10)*SBODY+K(11)                                            WTSCH
            GO TO 60                                                            WTSCH
         46 CONTINUE                                                            WTSCH
      C                                                                         WTSCH
      C     SAVE BODY VOLUME                                                    WTSCH
      C                                                                         WTSCH
            VBODYX=VBODY                                                        WTSCH
      C                                                                         WTSCH
      C     SUB TOTAL BODY VOLUME -STRUCTURE-MISC.                              WTSCH
      C                                                                         WTSCH
            VBODYA=VOXTK+VFUTK+VINSTK       +VPROP+VCARGO+VCREW+VLGBAY          WTSCH
           1+VFUTK2+VOXTK2                                                      WTSCH
```

```
C                                                                    WTSCH
C        COMPUTE BODY SURFACE AREA                                   WTSCH
C                                                                    WTSCH
      50 SBODY=CSBODY*VBODY2                                         WTSCH
C                                                                    WTSCH
C        COMPUTE STRUCTURAL VOLUME                                   WTSCH
C                                                                    WTSCH
         VSTRUC=K(10)*SBODY+K(11)                                    WTSCH
C                                                                    WTSCH
C                                                                    WTSCH
         VOTHER=K(18)*(VBODY-VCARGO-VSTRUC)+K(19)                    WTSCH
C                                                                    WTSCH
C        SUM BODY VOLUME                                             WTSCH
C                                                                    WTSCH
         VBODY=VBODYA+VSTRUC+VOTHER                                  WTSCH
C                                                                    WTSCH
         IF(VBODY.LE.0.0) VBODY=1.0                                  WTSCH
         IF(ABS(VBODYX-VBODY).LE.2.0) GO TO 60                       WTSCH
C                                                                    WTSCH
C        SAVE BODY VOLUME                                            WTSCH
C                                                                    WTSCH
         VBODYX=VBODY                                                WTSCH
C                                                                    WTSCH
C        UPDATE BODY VOLUME TO THE 2/3 POWER                         WTSCH
C                                                                    WTSCH
         IF(VBODY.LE.0.0) VBODY=1.0                                  WTSCH
         VBODY2=VBODY**C23                                           WTSCH
C                                                                    WTSCH
         GO TO 50                                                    WTSCH
C                                                                    WTSCH
C        UPDATE BODY VOLUME TO THE 1/3 POWER                         WTSCH
C                                                                    WTSCH
      60 VBODY1=VBODY**C13                                           WTSCH
C                                                                    WTSCH
C        UPDATE BODY VOLUME TO THE 2/3 POWER                         WTSCH
C                                                                    WTSCH
         VBODY2=VBODY1**2                                            WTSCH
C                                                                    WTSCH
         IF(NWL.LE.0.OR.NWL.GE.8) NWL=6                              WTSCH
C                                                                    WTSCH
C        TEST FOR WING SIZE METHOD-IF INPUT WING LOADING (FXWOVS) NOT EQUAL WTSCH
C        TO ZERO SIZE ON INPUT -IF INPUT  WING LOADING EQUAL ZERO USE INPUT WTSCH
C        WING AREA                                                   WTSCH
C                                                                    WTSCH
         IF(FXWOVS.NE.0.0) GO TO 70                                  WTSCH
C                                                                    WTSCH
C        VARIABLE WING LOADING                                       WTSCH
C                                                                    WTSCH
         WOVERS=0.0                                                  WTSCH
         IF (SWING.NE.0.0) WOVERS=WWAIT(NWL)/SWING                   WTSCH
C                                                                    WTSCH
         GO TO 80                                                    WTSCH
C                                                                    WTSCH
C        FIXED WING LOADING                                          WTSCH
C                                                                    WTSCH
      70 SWING=WWAIT(NWL)/FXWOVS                                     WTSCH
         WOVERS=FXWOVS                                               WTSCH
C                                                                    WTSCH
C        COMPUTE AERO WING SPAN                                      WTSCH
C                                                                    WTSCH
      80 GSPAN=0.0                                                   WTSCH
         IF((ASRATO*SWING).GT.0.0)                                   WTSCH
        1GSPAN=(ASRATO*SWING)**.5                                    WTSCH
C                                                                    WTSCH
C        COMPUTE WING ROOT CHORD                                     WTSCH
C                                                                    WTSCH
         CROOT=0.0                                                   WTSCH
         IF (TPRATO.NE.-1.0) CROOT=2.*SWING/((1.+TPRATO)*GSPAN)      WTSCH
C                                                                    WTSCH
C        COMPUTE WING TIP  CHORD                                     WTSCH
C                                                                    WTSCH
         CTIP=CROOT*TPRATO                                           WTSCH
C                                                                    WTSCH
```

```
C                                                                   WTSCH
C     COMPUTE WING STRUCTURAL SPAN 50PC                             WTSCH
C                                                                   WTSCH
      CSPAN=0.0                                                     WTSCH
      IF (GSPAN.NE.0.0)  CSPAN=GSPAN/COS(ATAN(TAN(ASWEEP/RTOD)-(.5*CROOT WTSCH
     1*(1.+TPRATO)/(GSPAN/2.))))                                    WTSCH
C                                                                   WTSCH
C     COMPUTE WING THICKNESS AT ROOT                                WTSCH
C                                                                   WTSCH
      TROOT=TOVERC*CROOT                                            WTSCH
C                                                                   WTSCH
C     COMPUTE BODY WIDTH                                            WTSCH
C                                                                   WTSCH
      BBODY=CBBODY*VBODY1                                           WTSCH
C                                                                   WTSCH
C     COMPUTE WING EXPOSED AREA                                     WTSCH
C                                                                   WTSCH
      SXPOS=0.0                                                     WTSCH
      IF((.5*BBODY).GT.0.0)                                         WTSCH
     1SXPOS =SWING-(CROOT*BBODY-(.5*BBODY)**2*TAN(ASWEEP/RTOD))     WTSCH
C                                                                   WTSCH
C     COMPUTE HORIZONTAL TAIL AREA                                  WTSCH
C                                                                   WTSCH
      SHORZ=CSHORZ*SWING                                            WTSCH
C                                                                   WTSCH
C     COMPUTE BODY SURFACE AREA                                     WTSCH
C                                                                   WTSCH
      SBODY=CSBODY*VBODY2                                           WTSCH
C                                                                   WTSCH
C     COMPUTE VERTICAL TAIL AREA                                    WTSCH
C                                                                   WTSCH
      SVERT=CSVERT*VBODY2                                           WTSCH
C                                                                   WTSCH
C     COMPUTE FAIRING AREA                                          WTSCH
C                                                                   WTSCH
      SFAIR=CSFAIR*SBODY                                            WTSCH
C                                                                   WTSCH
C     PLANFORM AREA                                                 WTSCH
C                                                                   WTSCH
      SPLAN=CSPLAN*VBODY2                                           WTSCH
C                                                                   WTSCH
C     COMPUTE BODY HEIGHT                                           WTSCH
C                                                                   WTSCH
      HBODY=CHBODY*VBODY1                                           WTSCH
C                                                                   WTSCH
C     COMPUTE BODY LENGTH                                           WTSCH
C                                                                   WTSCH
      LBODY=CLBODY*VBODY1                                           WTSCH
C                                                                   WTSCH
C                                                                   WTSCH
C     COMPUTE THERMAL PROTECTION AREA                               WTSCH
C                                                                   WTSCH
      IF(ITPS.LT.1.OR.ITPS.GT.8) ITPS=1                             WTSCH
      GO TO (90,100,110,120,130,140,150,160), ITPS                  WTSCH
   90 STPS(1)=0.                                                    WTSCH
      GO TO 170                                                     WTSCH
  100 STPS(1)=SBODY                                                 WTSCH
      GO TO 170                                                     WTSCH
  110 STPS(1)=SBODY+SHORZ                                           WTSCH
      GO TO 170                                                     WTSCH
  120 STPS(1)=SBODY+SHORZ+SVERT                                     WTSCH
      GO TO 170                                                     WTSCH
  130 STPS(1)=SBODY+SHORZ+SVERT+SWING                               WTSCH
      GO TO 170                                                     WTSCH
  140 STPS(1)=SHORZ+SVERT+SWING                                     WTSCH
      GO TO 170                                                     WTSCH
  150 STPS(1)=SBODY+SWING                                           WTSCH
      GO TO 170                                                     WTSCH
  160 STPS(1)=SBODY+SVERT                                           WTSCH
C                                                                   WTSCH
  170 CONTINUE                                                      WTSCH
C                                                                   WTSCH
```

```
C        **********                                                    WTSCH
C        WEIGHTS                                                       WTSCH
C        **********                                                    WTSCH
C                                                                      WTSCH
C                                                                      WTSCH
C        COMPUTE WING WEIGHT                                           WTSCH
C                                                                      WTSCH
         WWING=0.0                                                     WTSCH
         IF (TROOT.NE.0.0) WWING=C(1)*(WWAIT(6)* LF* CSPAN*SWING/TROOT WTSCH
        1/10.**9)**C(12)+C(2)*SWING+C(3)                               WTSCH
C                                                                      WTSCH
C        COMPUTE VERTICAL TAIL WEIGHT                                  WTSCH
C                                                                      WTSCH
         WVFRT=0.0                                                     WTSCH
         IF(SVERT.GT.0.0.AND.C(135).GT.0.0)                            WTSCH
        1WVERT=C(4)*SVERT**C(135)+C(24)*SVERT+C(5)                     WTSCH
C                                                                      WTSCH
C        COMPUTE HORIZONTAL TAIL WEIGHT                                WTSCH
C                                                                      WTSCH
         WHORZ=0.0                                                     WTSCH
         IF(WOVERS.GT.0.0.AND.SHORZ.GT.0.0.AND.Q.GT.0.0.AND.C(176).GT.0.0) WTSCH
        1WHORZ=C(6)*( WOVERS**1.21*SHORZ**.814*Q**.467)**C(176)+C(25)*SHORZ WTSCH
        2+C(7)                                                         WTSCH
C                                                                      WTSCH
C        COMPUTE FAIRING WEIGHT                                        WTSCH
C                                                                      WTSCH
         WFAIR=C(8)*SFAIR+C(9)                                         WTSCH
C                                                                      WTSCH
C        SUM AERO SURFACES                                             WTSCH
C                                                                      WTSCH
         WSURF=WWING+WVERT+WHORZ+WFAIR                                 WTSCH
C                                                                      WTSCH
C        COMPUTE INTEGRAL FUEL TANK WEIGHT                             WTSCH
C                                                                      WTSCH
         WINFUT=C(10)*VFUTK+C(11)                                      WTSCH
C                                                                      WTSCH
C        COMPUTE INTEGRAL LOX  TANK WEIGHT                             WTSCH
C                                                                      WTSCH
         WINOXT=C(138)*VOXTK+C(139)                                    WTSCH
C                                                                      WTSCH
C        COMPUTE BODY BASIC STRUCTURE WEIGHT                           WTSCH
C                                                                      WTSCH
         WBASIC=C(13)*SBODY+C(14)*VBODY+C(15)                          WTSCH
C                                                                      WTSCH
C        COMPUTE BODY SECONDARY STRUCTURE WEIGHT                       WTSCH
C                                                                      WTSCH
         WSECST=C(23)*SBODY+C(80)*WWAIT(7)+C(169)                      WTSCH
C                                                                      WTSCH
C        COMPUTE THRUST STRUCTURE WEIGHT                               WTSCH
C                                                                      WTSCH
         WTHRST=C(168)*TTOT+C(163)                                     WTSCH
C                                                                      WTSCH
C        SUM BODY WEIGHT                                               WTSCH
C                                                                      WTSCH
         WBODY=WINFUT+WINOXT+WBASIC+WSECST+WTHRST                      WTSCH
C                                                                      WTSCH
C        COMPUTE TPS INSULATION WEIGHT                                 WTSCH
C                                                                      WTSCH
         WINSUL=C(180)*STPS(1)+C(26)                                   WTSCH
C                                                                      WTSCH
C        COMPUTE TPS COVER PANELS WEIGHT                               WTSCH
C                                                                      WTSCH
         WCOVER=C(181)*STPS(1)+C(27)                                   WTSCH
C                                                                      WTSCH
C        SUM TPS WEIGHT                                                WTSCH
C                                                                      WTSCH
         WTPS=WINSUL+WCOVER                                            WTSCH
C                                                                      WTSCH
C        COMPUTE LAUNCH EQUIPMENT WEIGHT                               WTSCH
C                                                                      WTSCH
         WLANCH=C(143)*WTO+C(144)                                      WTSCH
C                                                                      WTSCH
C        COMPUTE LANDING GEAR WEIGHT                                   WTSCH
C                                                                      WTSCH
         WLG=0.0                                                       WTSCH
```

1092

```
      IF(WGROSS.GT.0.0.AND.C(182).GT.0.0)                                WTSCH
     1WLG=C(30)*WWAIT(7)**C(182)+C(31)                                   WTSCH
C                                                                        WTSCH
C     COMPUTE DOCKING EQUIPMENT                                          WTSCH
C                                                                        WTSCH
      WDOCK=C(147)*WWAIT(5)+C(148)                                       WTSCH
C                                                                        WTSCH
C     COMPUTE DEPLOYABLE AREO DEVICES                                    WTSCH
C                                                                        WTSCH
      WDPLOY=C(145)*WWAIT(7)+C(146)                                      WTSCH
C                                                                        WTSCH
C     SUM LAUNCH AND RECOVERY GEAR                                       WTSCH
C                                                                        WTSCH
      WLRD=WLANCH+WLG+WDOCK+WDPLOY                                       WTSCH
C                                                                        WTSCH
C     COMPUTE CREW WEIGHT                                                WTSCH
C                                                                        WTSCH
      WPERS=C(97)*NCREW+C(98)                                            WTSCH
C                                                                        WTSCH
C     COMPUTE ENGINE MOUNT WEIGHT                                        WTSCH
C                                                                        WTSCH
      WENGMT=C(183)*TTOT+C(184)                                          WTSCH
C                                                                        WTSCH
C     COMPUTE GIMBAL TORQUE                                              WTSCH
C                                                                        WTSCH
      TDEL=0.0                                                           WTSCH
      IF(NENGS.NE.0.0.AND.PCHAM.NE.0.0) TDEL=750.*(TTOT/NENGS/PCHAM)**   WTSCH
     1 1.25                                                              WTSCH
C                                                                        WTSCH
C     COMPUTE GIMBAL SYSTEM WEIGHT                                       WTSCH
C                                                                        WTSCH
      WSTAB=0.0                                                          WTSCH
      IF(TDEL.GT.0.0.AND.C(160).GT.0.0)                                  WTSCH
     1WSTAB=NENGS*(C(28)*TDEL**C(160))+C(161)                            WTSCH
C                                                                        WTSCH
C     COMPUTE SECONDARY ROCKET ENGINE WEIGHT                             WTSCH
C                                                                        WTSCH
      WENGS2=C(140)*TTOT2+C(141)                                         WTSCH
C                                                                        WTSCH
C     COMPUTE AIR BREATHING ENGINE WEIGHT                                WTSCH
C                                                                        WTSCH
      WABPR=C(210)*WWAIT(6)+C(211)                                       WTSCH
C                                                                        WTSCH
C     COMPUTE AIR BREATHING NACELLES WEIGHT                              WTSCH
C                                                                        WTSCH
      WNACEL=C(36)*WABPR+C(37)                                           WTSCH
C                                                                        WTSCH
C     TEST-FOR AIR BREATHING FUEL TYPE IF LH2 GO AROUND                  WTSCH
C       JP-4 SYSTEM WEIGHT EQUATION                                      WTSCH
C                                                                        WTSCH
      WBPUMP=0.0                                                         WTSCH
      WPRSYS=0.0                                                         WTSCH
      WDIST1=0.0                                                         WTSCH
      IF (C(212).NE.0.0.OR.C(213).NE.0.0) GO TO 180                      WTSCH  180
C                                                                        WTSCH
C     COMPUTE JP-4 FUEL PUMPS WEIGHT                                     WTSCH
C                                                                        WTSCH
      WBPUMP= C(149)*ANENGS*(1.75+.266*ANENGS)/1000.                     WTSCH
C                                                                        WTSCH
C     COMPUTE JP-4 PRESSURE SYSTEM WEIGHT                                WTSCH
C                                                                        WTSCH
      WABFPS=0.0009*C(149)*ANENGS*ANTANK                                 WTSCH
C                                                                        WTSCH
C     COMPUTE JP-4 FUEL DISTRIBUTION SYSTEM WEIGHT                       WTSCH
C                                                                        WTSCH
      WDIST1=ANENGS*C(191)*SQRT(C(149))                                  WTSCH
C                                                                        WTSCH
  180 CONTINUE                                                           WTSCH
C                                                                        WTSCH
C     COMPUTE ROCKET ENGINE WEIGHT                                       WTSCH
C                                                                        WTSCH
      WENGS=0.0                                                          WTSCH
      IF(C(221).GT.0.0)                                                  WTSCH
     1WENGS=C(32)*TTOT+C(219)*TTOT*C(220)**C(221)+C(33)*NENGS+WENGMT     WTSCH
C                                                                        WTSCH
```

1602

```
C      COMPUTE ROCKET FUEL TANK WEIGHT NON-STRUCTURAL                     WTSCH
C                                                                         WTSCH
       WFUTK=C(39)*VFUTK+C(40)                                            WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET LOX  TANK WEIGHT NON-STRUCTURAL                     WTSCH
C                                                                         WTSCH
       WOXTK=C(41)*VOXTK+C(42)                                            WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET SECONDARY FUEL TANK WEIGHT                          WTSCH
C                                                                         WTSCH
       WFUTK2=C(170)*VFUTK2+C(136)                                        WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET SECONDARY LOX  TANK WEIGHT                          WTSCH
C                                                                         WTSCH
       WOXTK2=C(171)*VOXTK2+C(137)                                        WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET PROPELLANT TANK INSULATION WEIGHT                   WTSCH
C                                                                         WTSCH
       WINSTK=C(43)*SFUTK+C(77)*SOXTK+C(44)                               WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET FUEL SYSTEM WEIGHT                                  WTSCH
C                                                                         WTSCH
       WFUSYS=C(45)*TTOT+C(46)*LBODY+C(47)                                WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET LOX  SYSTEM WEIGHT                                  WTSCH
C                                                                         WTSCH
       WOXSYS=C(48)*TTOT+C(49)*LBODY+C(50)                                WTSCH
C                                                                         WTSCH
C      COMPUTE ROCKET PRESSURE SYSTEM WEIGHT                              WTSCH
C                                                                         WTSCH
       WPRSYS=C(51)*VFUTK+C(52)*VOXTK+C(187)                              WTSCH
C                                                                         WTSCH
C      TEST-FOR AIR BREATHING FUEL TYPE IF LH2 GO AROUND                  WTSCH
C        JP-4 SYSTEM WEIGHT EQUATION                                      WTSCH
C                                                                         WTSCH
       GAL=0.0                                                            WTSCH
       WFUNCT=0.0                                                         WTSCH
       WDIST2=0.0                                                         WTSCH
       WFCONT=0.0                                                         WTSCH
       WREFUL=0.0                                                         WTSCH
       WDRANS=0.0                                                         WTSCH
       WSEAL=0.0                                                          WTSCH
       ABFSYS=0.0                                                         WTSCH
       WABFS = 0.                                                         WTSCH
       IF (C(212).NE.0.0.OR.C(213).NE.0.0) GO TO 190                      WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL IN GALLONS                                       WTSCH
C                                                                         WTSCH
       GAL=WABFU/6.5                                                      WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL SYSTEM TANKS WEIGHT                              WTSCH
C                                                                         WTSCH
       IF(ANTANK.NE.0.0) WFUNCT=C(189)*(GAL/ANTANK)**.6*ANTANK+C(190)     WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL SYSTEM DISTRIBUTION WEIGHT                       WTSCH
C                                                                         WTSCH
       WDIST2=.255*GAL**.7*ANTANK**.25                                    WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL SYSTEM CONTROLS WEIGHT                           WTSCH
C                                                                         WTSCH
       WFCONT=.169*ANTANK*SQRT(GAL)                                       WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL SYSTEM REFUEL SYSTEM WEIGHT                      WTSCH
C                                                                         WTSCH
       WREFUL=ANTANK*(3.+.45*GAL**C13)                                    WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL SYSTEM REFUEL SYSTEM WEIGH                       WTSCH
C                                                                         WTSCH
       WDRANS=.159*GAL**.65                                               WTSCH
C                                                                         WTSCH
C      COMPUTE JP-4 FUEL TANK SEALING                                     WTSCH
C                                                                         WTSCH
       IF(ANTANK.NE.0.0) WSEAL=.045*ANTANK*(GAL/ANTANK)**.75              WTSCH
C                                                                         WTSCH
C      SUM JP-4 FUEL SYSTEM WEIGHT                                        WTSCH
```

190

1603

```
      C                                                                   WTSCH
             ABFSYS=WBPUMP+WDIST1+WDIST2+WFCONT+WREFUL+WDRANS+WSEAL+WFUNCT WTSCH
            1+WABFPS                                                      WTSCH
      C                                                                   WTSCH
      C      SUM JP FUEL SYSTEM WEIGHT LESS TANKS                         WTSCH
      C                                                                   WTSCH
             WABFS=ABFSYS-WFUNCT                                          WTSCH
      C                                                                   WTSCH
         190 CONTINUE                                                     WTSCH
      C                                                                   WTSCH
      C      COMPUTE AIRFUEL TANK WEIGHT                                  WTSCH
      C                                                                   WTSCH
             WABFTK=C(212)*WABFU+C(213) +ABFSYS                           WTSCH
      C                                                                   WTSCH
      C      SUM PROPULSION SYSTEM WEIGHT                                 WTSCH
      C                                                                   WTSCH
             WPROP=WENGS+WNACEL+WFUTK+WOXTK+WINSTK+WFUSYS+WOXSYS+WPRSYS+WENGS2+ WTSCH
            1WFUTK2+WOXTK2+WABPR+WABFTK                                   WTSCH
      C                                                                   WTSCH
      C      COMPUTE AREO SURFACE CONTROL WEIGHT                          WTSCH
      C                                                                   WTSCH
             WAERO=0.0                                                    WTSCH
             IF(WWAIT(5).GT.0.0.AND.(LBODY+CSPAN).GT.0.0.AND.C(185).GT.0.0) WTSCH
            1WAERO=C(55)*(WWAIT(5)**.689*(LBODY+CSPAN)**.287)**C(185)*C(56) WTSCH
      C                                                                   WTSCH
      C      COMPUTE SEPARATION SYSTEM WEIGHT                             WTSCH
      C                                                                   WTSCH
             IF(JUMP.EQ.0.0) WAUXT=C(153)*WTO+C(154)                      WTSCH
             IF(JUMP.EQ.1.0) WAUXT=C(153)*WPAYL+C(154)                    WTSCH
      C                                                                   WTSCH
      C      COMPUTE ACS SYSTEM WEIGHT                                    WTSCH
      C                                                                   WTSCH
             WACS=0.0                                                     WTSCH
             IF(WTO.GT.0.0.AND.C(155).GT.0.0)                             WTSCH
            1WACS=C(156)*WWAIT(4)**C(155)*C(157)                          WTSCH
      C                                                                   WTSCH
      C      COMPUTE ACS PROPELLANT TANK WEIGHT                           WTSCH
      C                                                                   WTSCH
             WACSTK=C(164)*(WACSFO+WACRES)+C(165)                         WTSCH
      C                                                                   WTSCH
      C      SUM ORIENTATION CONTROLS WEIGHT                              WTSCH
      C                                                                   WTSCH
             WORSUL=WSTAB+WACS+WAERO+WAUXT+WACSTK                         WTSCH
      C                                                                   WTSCH
      C      COMPUTE GUIDANCE AND NAVIGATION SYSTEM WEIGHT                WTSCH
      C                                                                   WTSCH
             WGNAV=C(68)                                                  WTSCH
      C                                                                   WTSCH
      C      COMPUTE INSTRUMENTATION SYSTEM WEIGHT                        WTSCH
      C                                                                   WTSCH
             WINST =C(69)*LBODY+C(70)                                     WTSCH
      C                                                                   WTSCH
      C      COMPUTE COMMUNICATION SYSTEM WEIGHT                          WTSCH
      C                                                                   WTSCH
             WCOMM=C(71)*NCREW+C(72)                                      WTSCH
      C                                                                   WTSCH
      C      SUM AVIONICS SYSTEM                                          WTSCH
      C                                                                   WTSCH
             WAVIOC=WGNAV+WINST +WCOMM                                    WTSCH
      C                                                                   WTSCH
      C      COMPUTE PRIME POWER SOURCE AND DISTRIBUTION                  WTSCH
      C                                                                   WTSCH
             WSORCE=C(62)  *  WWAIT(6)  +  C(64)                          WTSCH
      C                                                                   WTSCH
      C      COMPUTE PRIME POWER SOURCE TANKAGE                           WTSCH
      C                                                                   WTSCH
             WPOWTK=C(29)*WPOWFO+C(60)                                    WTSCH
      C                                                                   WTSCH
      C      SUM ELECTRICAL POWER SYSTEMS                                 WTSCH
      C                                                                   WTSCH
             WPOWER=WSORCE+WPOWTK                                         WTSCH
      C                                                                   WTSCH
      C      COMPUTE HYDRAULIC/PNEUMATIC SYSTEM                           WTSCH
      C                                                                   WTSCH
             WHYCAD=0.0                                                   WTSCH
```

```
      IF(((SWING+SHORZ+SVERT)*Q/1000.).GT.0.0.AND.(LBODY+CSPAN).GT.0.0)  WTSCH
     1WHYCAD=C(65)*(((SWING+SHORZ+SVERT)*Q/1000.)**1.3125*(LBODY+CSPAN)  WTSCH
     2**1.06125)**C(66)+C(67)                                            WTSCH
C                                                                        WTSCH
C     SUM POWER CONTROL AND DISTRIBUTION                                 WTSCH
C                                                                        WTSCH
      WPOWCD=WHYCAD                                                      WTSCH
C                                                                        WTSCH
C     COMPUTE PERSONNEL ACCOMODATIONS WEIGHT                             WTSCH
C                                                                        WTSCH
      WPPROV=C(74)*WWAIT(7)+C(75)*NCREW+C(76)                            WTSCH
C                                                                        WTSCH
C     COMPUTE DRY WEIGHT                                                 WTSCH
C                                                                        WTSCH
      WDRY=WSURF+WBODY+WTPS+WLRD+WPROP+WORSUL+WPOWCD+WGNAV+WINST +WCOMM+ WTSCH
     1WPPROV +WPOWER                                                     WTSCH
C                                                                        WTSCH
C     COMPUTE GROWTH AND CONTINGENCY                                     WTSCH
C                                                                        WTSCH
      CRAP = WDRY-WENGS-WABPR-C(141)                                     WTSCH
      WCONT = C(96)*CRAP + C(162)                                        WTSCH
C                                                                        WTSCH
C     SUM WEIGHT WEIGHT EMPTY                                            WTSCH
C                                                                        WTSCH
      WEMPTY=WDRY+WCONT                                                  WTSCH
C                                                                        WTSCH
C     COMPUTE CARGO WEIGHT                                               WTSCH
C                                                                        WTSCH
      WCARGO=C(102)*NPASS+C(103)                                         WTSCH
C                                                                        WTSCH
C     COMPUTE PASSENGER WEIGHT                                           WTSCH
C                                                                        WTSCH
      WPASS=C(104)*NPASS+C(105)                                          WTSCH
C                                                                        WTSCH
C     SUM PAYLOAD WEIGHT                                                 WTSCH
C                                                                        WTSCH
      WPAYL=WPASS+WCARGO                                                 WTSCH
C                                                                        WTSCH
C     SUM OPERATING WEIGHT EMPTY                                         WTSCH
C                                                                        WTSCH
      WWET  =WEMPTY+WRESID+WPERS+WPAYL+WRESRV                            WTSCH
C                                                                        WTSCH
C     SUM ZERO FUEL WEIGHT                                               WTSCH
C                                                                        WTSCH
      WZROFU=   WWET+      WFULOS+WOXLOS                                 WTSCH
     1                             +WACSFO      +WPOWFO+WOIL+WABFU+WFROST WTSCH
     2 + C(296)                                                          WTSCH
C                                                                        WTSCH
C     SUM TAKEOFF WEIGHT                                                 WTSCH
C                                                                        WTSCH
      WTO=WZROFU+WFUOX+WDECAY-WFUEL(1)-WOX(1)                            WTSCH
C                                                                        WTSCH
C     SUM GROSS WEIGHT                                                   WTSCH
C                                                                        WTSCH
      WGROSS=WTO+WPREIG+WFUEL(1)+WOX(1)                                  WTSCH
C                                                                        WTSCH
C     UPDATE WWAIT(1)                                                    WTSCH
C                                                                        WTSCH
      WWAIT(1)=WGROSS -WPREIG                                            WTSCH
C                                                                        WTSCH
C     GROSS WEIGHT LESS PAYLOAD                                          WTSCH
C                                                                        WTSCH
      WTABC=WGROSS-WPAYL                                                 WTSCH
C                                                                        WTSCH
C                                                                        WTSCH
C     MISCELLANEOUS CONTROL STATEMENTS                                   WTSCH
C                                                                        WTSCH
C                                                                        WTSCH
      IF(NLISTO.NE.0) WRITE(6,MASS1)                                     WTSCH
      IF(NLISTO.NE.0) WRITE(6,MASS2)                                     WTSCH
C                                                                        WTSCH
      IF (WGROSS.GT.20000000..AND.JUMP.EQ.1.OR.WGROSS.GT.10000000..AND.J WTSCH   210
     1UMP.EQ.0) GO TO 210                                                WTSCH
C                                                                        WTSCH
      RETURN                                                             WTSCH
```

```
C                                                                   WTSCH
  210 WRITE(6,MASS1)                                                WTSCH
      WRITE(6,MASS2)                                                WTSCH
C                                                                   WTSCH
      CALL EXIT                                                     WTSCH
C                                                                   WTSCH
      END                                                           WTSCH
```

SUBRØUTINE WTVØL

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
BTTOT		O	Total booster weight flow	/TAMP /(	5)	SUMOUT	I	BTTOT
						TAMPER	I	BTTOT
						WTVOL	O	BTTOT
C		O	Input array c(300) of vehicle sizing data	/CINPUT/(	5)	PRINTW	I	C
						PRITED	I	C
						PRITVA	I	C
						STORE	M	C
						WTSCH	I	C
						WTVOL	O	C
CTHRST		M	Vac. Thrust-to-weight ratio	/CINPUT/(	322)	PRITVA	I	CTHRST
						STORE	M	CTHRST
						WTSCH	I	CTHRST
						WTVOL	M	CTHRST
ILAST		W	Print cycle flag 1 = print cycle 0 = not on a print cycle	/WTVOL /(*	)	WTVOL	W	ILAST
ISP		O	Specific impulse	/CINPUT/(	330)	PRWTSM	I	ISP
						STORE	M	ISP
						WTSCH	I	ISP
						WTVOL	O	ISP
JUMP		M	Data flag 0= orbiter 1= booster	/JUMPY /(	1)	FRENCH	O	JUMP
						PRINTW	I	JUMP
						PRITVA	I	JUMP
						PRWTSM	M	JUMP
						WTSCH	I	JUMP
						WTVOL	M	JUMP
OTTOT		M	Total orbiter thrust	/TAMP /(	2)	SUMOUT	I	OTTOT
						TAMPER	I	OTTOT
						WTVOL	M	OTTOT
PRINTV		S	Subroutine to print vehicle volume data	/PRINTV/($	)	PRINTV	E	PRINTV
						WTVOL	S	PRINTV
PRINTW		S	Subroutine to print vehicle weight data	/PRINTW/($	)	PRINTW	E	PRINTW
						WTVOL	S	PRINTW
PRWTSM		S	Subroutine to print summary data	/PRWTSM/($	)	PRWTSM	E	PRWTSM
						WTVOL	S	PRWTSM
RPAR		W	Weighting factor for updating range	/WTVOL /(*	)	WTVOL	W	RPAR
SCB		M	Working name for input c-array booster scaling coefficients	/ORBINV/(	144)	FLYBKP	M	SCB
						STORE	M	SCB
						SUMOUT	I	SCB
						TAMPER	I	SCB
						THRUST	M	SCB
						VEHDF	I	SCB
						WTVOL	M	SCB
SE		M	Array of synthesis iteration propulsion parameters	/SIZING/(	259)	FLYBKP	O	SE
						PRITVA	I	SE
						SIZEMR	I	SE
						SUMOUT	I	SE
						TAMPER	M	SE
						THRUST	I	SE
						VEHDF	M	SE
						WTSCH	I	SE
						WTVOL	M	SE
SOLVE		S	Subroutine to drive weight and volume calculations (wtsch) to convergence - an inner loop driver	/SOLVE /($	)	SOLVE	E	SOLVE
						WTVOL	S	SOLVE
SPLAN		I	Body planform area	/VOLCAL/(	14)	PROTHR	I	SPLAN
						TAMPER	I	SPLAN
						WTSCH	O	SPLAN
						WTVOL	I	SPLAN

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
SQ		M	A synthesis data array (37,5) that contains the flyback data and some injection quantities	/SIZING/(	74)	ENVPRM	M	SQ
						FLYBKP	M	SQ
						ISPRAT	I	SQ
						PDBC	I	SQ
						PRITVA	I	SQ
						RANGE	M	SQ
						REU3	O	SQ
						SIZE	O	SQ
						SIZEMR	M	SQ
						SIZIN	M	SQ
						STAU	I	SQ
						SUMOUT	M	SQ
						TAMPAR	O	SQ
						TAMPER	M	SQ
						THRUST	M	SQ
						TRTOSZ	M	SQ
						VEHDF	M	SQ
						WTVOL	M	SQ
SV		I	A synthesis array (28) containing staging parameters and misc flags	/SIZING/(	46)	ENVPRM	M	SV
						FLYBKP	I	SV
						ITER8	I	SV
						RANGE	I	SV
						SIZEMR	M	SV
						SIZIN	I	SV
						SSSP	I	SV
						SUMOUT	I	SV
						TAMPAR	O	SV
						TAMPER	M	SV
						TRTOSZ	M	SV
						VEHDF	M	SV
						WTVOL	I	SV
TAMPER		S	Subroutine to interface weight and volume overlay with trajectory program overlay	/TAMPER/(S	)	TAMPER	E	TAMPER
						WTVOL	S	TAMPER
TB18		O	Stored booster value of cthrst	/ORBINY/(	23)	STORE	M	TB18
						WTVOL	O	TB18
TB27		I	Stored booster value of isp(i)	/ORBINY/(	41)	SIZEMR	I	TB27
						SSSP	I	TB27
						STORE	M	TB27
						SUMOUT	I	TB27
						TAMPER	I	TB27
						VEHDF	M	TB27
						WTVOL	I	TB27
TB34		M	Stored booster value of mr(i)	/ORBINY/(	53)	FLYBKP	I	TB34
						ITER8	O	TB34
						SSSP	M	TB34
						STORE	M	TB34
						SUMOUT	I	TB34
						TAMPER	I	TB34
						VEHDF	I	TB34
						WTVOL	M	TB34
TB36		I	Stored booster value of nengs	/ORBINY/(	60)	STORE	M	TB36
						SUMOUT	I	TB36
						TAMPER	I	TB36
						THRUST	I	TB36
						WTVOL	I	TB36
TB51		O	Stored booster value of vbody	/ORBINY/(	102)	STORE	M	TB51
						WTVOL	O	TB51
TB56		M	Stored booster value of egress	/ORBINY/(	107)	STORE	M	TB56
						WTVOL	M	TB56
TO18		M	Stored orbiter value of cthrst	/ORBINX/(	23)	STORE	M	TO18
						WTVOL	M	TO18
TO27		I	Stored orbiter value of isp(i)	/ORBINX/(	41)	SIZEMR	I	TO27
						SSSP	I	TO27
						STORE	M	TO27
						SUMOUT	I	TO27
						VEHDF	M	TO27
						WTVOL	I	TO27

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	CODE	VAR
T034		M	Stored orbiter value of mr(1)	/ORBINX/(	53)	ITER8	M	T034
						SSSP	O	T034
						STORE	M	T034
						TAMPER	I	T034
						VEHOF	O	T034
						WTVOL	M	T034
T036		I	Stored orbiter value of mengs	/ORBINX/(	60)	STORE	M	T036
						SUMOUT	I	T036
						TAMPER	I	T036
						THRUST	I	T036
						WTVOL	I	T036
T051		O	Stored orbiter value of vbody	/ORBINX/(	102)	STORE	M	T051
						WTVOL	O	T051
T056		M	Stored orbiter value of mgross	/ORBINX/(	107)	STORE	M	T056
						WTVOL	M	T056
TTOT		I	Total stage vac. Thrust	/VOLCAL/(	21)	PRITVA	I	TTOT
						STORE	I	TTOT
						WTSCH	M	TTOT
						WTVOL	I	TTOT
TWO		W	Thrust-to-weight ratio	/WTVOL /(*	)	WTVOL	W	TWO
VBODY		I	Total body volume	/CINPUT/(	391)	PRINTV	M	VBODY
						SOLVE	M	VBODY
						STORE	M	VBODY
						TAMPER	I	VBODY
						WTSCH	M	VBODY
						WTVOL	I	VBODY
VBODY1		M	Vbody to - 1/3 power	/VOLCAL/(	25)	WTSCH	M	VBODY1
						WTVOL	M	VBODY1
VBODY2		M	Vbody to - 2/3 power	/VOLCAL/(	26)	WTSCH	M	VBODY2
						WTVOL	M	VBODY2
WBOO		O	Booster gross weight	/JUMPY /(	3)	PRINTW	I	WBOO
						TAMPER	I	WBOO
						WTVOL	O	WBOO
WFUOX		I	Weight of main and secondary propellant	/WTCALC/(	46)	PRINTW	I	WFUOX
						TAMPER	I	WFUOX
						WTSCH	M	WFUOX
						WTVOL	I	WFUOX
WFUOXO		M	Propellant wt. Less fpr -orbiter	/TAMP /(	3)	SUMOUT	I	WFUOXO
						TAMPER	I	WFUOXO
						WTVOL	M	WFUOXO
WGROSO		M	Orbiter gross weight	/TAMP /(	1)	SSSP	I	WGROSO
						SUMOUT	I	WGROSO
						WTVOL	M	WGROSO
WGROSS		I	Gross lift-off weight	/CINPUT/(	392)	PRINTW	I	WGROSS
						PRWTSM	I	WGROSS
						SOLVE	M	WGROSS
						STORE	M	WGROSS
						TAMPER	I	WGROSS
						WTSCH	M	WGROSS
						WTVOL	I	WGROSS
WOREQ		W	Required orbiter gross weight	/WTVOL /(*	)	WTVOL	W	WOREQ
WPOREQ		W	Required orbiter propellant weight	/WTVOL /(*	)	WTVOL	W	WPOREQ
WTSCH		S	Subroutine to calculate weight and volume of both stages	/WTSCH /($	)	SOLVE	S	WTSCH
						WTSCH	E	WTSCH
						WTVOL	S	WTSCH
WTVOL		E	Main program for weight and volume calculations overlay(5,2)	/WTVOL /($	)	WTVOL	E	WTVOL

FORTRAN SYMBOL	MATH SYMBOL	CODE	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBROUTINE USAGE CODE	SUBROUTINE USAGE VAR
.UN06.		0	File of all output data	/.UN06./(6	)	BLICO	0	.UN06.
						BNDRYC	0	.UN06.
						CRASH	0	.UN06.
						FRENCH	0	.UN06.
						FXDAT	0	.UN06.
						GEINP	0	.UN06.
						HUNT	0	.UN06.
						INEDIT	0	.UN06.
						ITER8	0	.UN06.
						MODELA	0	.UN06.
						MOMJ	0	.UN06.
						MPSI	0	.UN06.
						OUT	0	.UN06.
						PAYO2	0	.UN06.
						PRINT	0	.UN06.
						PRINTV	0	.UN06.
						PRINTW	0	.UN06.
						PRITEQ	0	.UN06.
						PRITVA	0	.UN06.
						PROPIN	0	.UN06.
						PROTHR	0	.UN06.
						PRWTSM	0	.UN06.
						RANGE	0	.UN06.
						S	0	.UN06.
						SDINP	0	.UN06.
						SIZE	0	.UN06.
						SIZIN	0	.UN06.
						SIZOUT	0	.UN06.
						SOLVE	0	.UN06.
						SPLICO	0	.UN06.
						SPLIZ	0	.UN06.
						SPLYNE	0	.UN06.
						SSSP	0	.UN06.
						STAU	0	.UN06.
						STPIT	0	.UN06.
						SUMOUT	0	.UN06.
						TABIN	0	.UN06.
						TEST	0	.UN06.
						VEHDF	0	.UN06.
						WTSCH	0	.UN06.
						WTVOL	0	.UN06.

```
      PROGRAM WTVOL                                                     WTVOL
C**********************************************************************  WTVOL
C            PRIMARY SUBROUTINE FOR WEIGHT AND VOLUME SUBROUTINE        WTVOL
C                                                                       WTVOL
      REAL MUB, MUO, ISPB, ISPO, IDVEL,NNB,NO                           SIZING
      COMMON /SIZING/                                                   SIZING
C          PHASE II SIZING PARAMERERS                                   SIZING
     *TZ,      VV(3),      OP(14),    EROR,     PZ(5),      VQ,       SW(20), SIZING
     *SV(28),  SQ(37,5),   SE(11),    TLAT,     TLNG,                   SIZING
C         PHASE I SIZING PARAMERERS                                     SIZING
     *WBO,     WLOO,       DWEB,      DWEO,     TOLWT,     WPB,      TWRAT2, SIZING
     *BK1,     BK2,        BK3,       BK4,      ISIZE,     TRAFLG,   TWRATO, SIZING
     *DK1,     OK2,        OK3,       OK4,      PRFLG,     IPASS,    IPSMAX, SIZING
     *AEXIT,   TVACO,      NO,        WFO,      IDVEL,     ISPO,     ISPB,  SIZING
     *XPL,     TVACB,   NNB,          WEO,      WEB,       WO,       WLO,   SIZING
     *DVO,      DVB,        MUB,       MUO,      VSTG,      WPO             SIZING
     *,JTYP,    BECO       BSTG       ORBI      ITNBW   , ITNOW   ,        SIZING
     * SVOPSQ  , SVDCON   ,IHUNT     ,IOPSTG   ,ISZO(19)                    UH
      DIMENSION SKO(30),SCO(300),TO4(6),TO20(10),TO27(6),TO34(6),       ORBINX
     1 TO48(10),TO49(10),TO50(10),TO57(6),OWSAVE(10)                   CKOUT
      COMMON/ORBINX/                                                    ORBINX
     1 TO1,TO2,TO3,TO4,TO5,TO6,TO7,TO8,TO9,TO10,TO11,TO12,TO13,TO14,   ORBINX
     2 TO15,TO16,TO17,TO18,TO19,TO20,TO21,IO22,IO23,IO24,IO25,IO26,TO27, ORBINX
     3 IO28,IO29,IO30,IO31,IO32,TO33,TO34,TO35,TO36,IO37,TO38,TO39,TO40, ORBINX
     4 TO41,TO42,TO43,TO44,TO45,TO46,TO47,TO48,TO49,TO50,TO51,TO52,TO53, ORBINX
     5 TO54, TO55,TO56,TO57,TO66,SKO.SCO,OWSAVE                          CKOUT
     6,TO59,TO60,TO61,TO62,TO63,TO64,TO65,TO67,TO68,TO69,TO70,TO71,     UH
     7 TO72,TO73,TO74,TO75,TO76,TO77,TO78,TO79,TO80,TO81,TO82,TO83,     UH
     8TO84                                                              UH
      DIMENSION SKB(30),SCB(300),TB4(6),TB20(10),TB27(6),TB34(6),       ORBINY
     1 TB48(10),TB49(10),TB50(10),TB57(6),BWSAVE(10)                   CKOUT
      COMMON/ORBINY/                                                    ORBINY
     1 TB1,TB2,TB3,TB4,TB5,TB6,TB7,TB8,TB9,TB10,TB11,TB12,TB13,TB14,   ORBINY
     2 TB15,TB16,TB17,TB18,TB19,TB20,TB21,IB22,IB23,IB24,IB25,IB26,TB27, ORBINY
     3 IB28,IB29,IB30,IB31,IB32,TB33,TB34,TB35,TB36,IB37,TB38,TB39,TB40, ORBINY
     4 TB41,TB42,TB43,TB44,TB45,TB46,TB47,TB48,TB49,TB50,TB51,TB52,TB53, ORBINY
     5 TB54,TB55,TB56,TB57,SKB,SCB,BWSAVE                                UH
     6,TB59,TB60,TB61,TB62,TB63,TB64,TB65,TB66,TB67,TB68,TB69,TB70,     UH
     7TB71,TB72,TB73,TB74,TB75,TB76,TB77,TB78,TB79,TB80,TB81,TB82,      UH
     8TB83,TB84                                                         UH
      REAL KIN                                                          CINPUT
      REAL ISP,K,LF,MR,NCREW,LBODY,NPASS                                CINPUT
      REAL NENGS                                                        CINPUT
      COMMON/CINPUT/                                                    CINPUT
     1ANENGS  ,ANTANK  ,ASRATO  ,ASWEEP  ,C(300)  ,CBBODY  ,CFUEL(6),   CINPUT
     2CHBODY  ,CLBODY  ,CSBODY  ,CSFAIR  ,CSFUTK  ,CSHORZ  ,CSOXTK  ,   CINPUT
     3CSPLAN  ,CSVERT  ,CSWING  ,CTHRST  ,CTHST2  ,DEF(5)  ,FXWOVS  ,   CINPUT
     4ISP(6)  ,ITPS    ,K(30)   .KIN     ,LF      ,MR(6)   ,NCREW   ,   CINPUT
     5NENGS   ,NLISTO  ,NPASS   ,NWL     ,PCHAM   ,Q       ,RHOFU   ,   CINPUT
     6RHOFU2  ,RHOX    ,RHOX2   ,SBODY   ,TOL     ,TOVERC  ,TPRATO  ,   CINPUT
     7TYTAIL  ,VBODY   ,WGROSS                                         CINPUT
      COMMON/VOLCAL/BBODY,CROOT ,CSPAN   ,CTIP    ,GAL     ,GSPAN   ,   VOLCAL
     2HBODY   ,LBODY   ,RTOD    ,SFAIR   ,SFUTK   ,SHORZ   ,SOXTK   ,   VOLCAL
     3SPLAN   ,STPS(1) ,SVERT   ,SWING   ,SXPOS   ,TDEL    ,TROOT   ,   VOLCAL
     4TTOT    ,TTOT2   ,TTOTAL  ,VBODYA  ,VBODY1  ,VBODY2  ,VCARGO  ,   VOLCAL
     5VCREW   ,VFUTK   ,VFUTK2  ,VINSTK  ,VLGBAY  ,VOTHER  ,VOXTK   ,   VOLCAL
     6VOXTK2  ,VPROP   ,VSTRUC                                         VOLCAL
      COMMON/WTCALC/    ABFSYS  ,WABFTK  ,WABFU   ,WABPR   ,WACRES  ,   WTCALC
     1WACS    ,WACSFO  ,WACSTK  ,WAERO   ,WAUXT   ,WBASIC  ,WBODY   ,   WTCALC
     2WBPUMP  ,WCARGO  ,WCOMM   ,WCONT   ,WCOVER  ,WDECAY  ,WDIST1  ,   WTCALC
     3WDIST2  ,WDOCK   ,WDPLOY  ,WDRANS  ,WDRY    ,WELCAD  ,WEMPTY  ,   WTCALC
     4WENGMT  ,WENGS   ,WENGS2  ,WFAIR   ,WFCONT  ,WFDCAY  ,WFROST  ,   WTCALC
     5WFU2(3) ,WFUEL(6),WFUL    ,WFULOS  ,WFUNCT  ,WFUOX   ,WFURES  ,   WTCALC
     6WFUSYS  ,WFUTK   ,WFUTK2  ,WFUTOT  ,WFUTRP  ,WGASPR  ,WGNAV   ,   WTCALC
     7WHORZ   ,WHYCAD  ,WINFUT  ,WINOXT  ,WINSTK  ,WINST   ,WINSUL  ,   WTCALC
     8WJET(6) ,WLANCH  ,WLG     ,WLOSS   ,WLRD    ,WNACEL  ,WODCAY  ,   WTCALC
     9WOIL    ,WOILRS  ,WORSUL  ,WOVERS  ,WOX(6)  ,WOX2(3) ,WOXID   ,   WTCALC
     1WOXLOS  ,WOXRES  ,WOXSYS  ,WOXTK   ,WOXTK2  ,WOXTOT  ,WOXTRP  ,   WTCALC
     2WP      ,WPASS   ,WPAYL   ,WPERS   ,WPOWCO  ,WPOWER  ,WPOWFO  ,   WTCALC
     3WPOWRS  ,WPOWTK  ,WPPROV  ,WPREIG  ,WPROP   ,WPRSYS  ,WREFUL  ,   WTCALC
     4WRESIO  ,WRESRV  ,WSEAL   ,WSECST  ,WSORCE  ,WSRTRP  ,WSTAB   ,   WTCALC
     5WSURF   ,WTABC   ,WTHRST  ,WTO     ,WTPS    ,WVERT   ,WWAIT(10),  WTCALC
     6WWET    ,WWING   ,WZROFU  ,WABTRP  ,WABRES  ,WMNOTP  ,WMNFTP  ,   CKOUT
     7WMNORS  ,WMNFRS  ,WACOTP  ,WACFTP  ,WPWOTP  ,WPWFTP  ,WGAS    ,   CKOUT
     8WABFUC  ,WACORS  ,WACFRS  ,WPWORS  ,WPWFRS                       CKOUT
```

```
C                                                                    WTVOL
      COMMON/IAMP/WGROSO,OTTOT,WFUOXO,TBTO,BTTOT                     WTVOL
      COMMON/JUMPY/ JUMP, WBIG, WBOD                                 CKOUT
      COMMON/TRUST/                                                  CKOUT
     * FVACO,    FSLO,     FVACLO,     FVACS,                        CKOUT
     * FVACB,    FSLB,    FSLLB,    FSLS                             CKOUT
C                                                                    WTVOL
C          CHECK PRINT FLAG                                          WTVOL
C                                                                    WTVOL
      BTWTOL=SW(18)*.5                                               WTVOL
      QMX = SE(6)                                                    WTVOL
      WOREQ= SQ(10,1)                                                WTVOL
      WPOREQ = SQ(13,1)                                              WTVOL
      GWREQ= SQ(16,1)                                                WTVOL
      IF(SW(3).GE.1.) GO TO 40                                       WTVOL
C                                                                    WTVOL
C          FIRST PASS LOGIC                                          WTVOL
C                                                                    WTVOL
      SQ(15,2)=1.                                                    WTVOL
C                                                                    WTVOL
C          NORMAL ENTRY POINT - INITALIZE VARIABLES                  WTVOL
C                                                                    WTVOL
   40 CONTINUE                                                       WTVOL
      SW(3)=SW(3)+1.                                                 WTVOL
      BIB=.00005                                                     WTVOL
      BOB=.00025                                                     WTVOL
      ILAST=0                                                        WTVOL
      EROR  = 0.1                                                    WTVOL
      GUS = 0.                                                       WTVOL
      ACEL=0.                                                        WTVOL
      ACNT=0.                                                        WTVOL
  101 CONTINUE                                                       WTVOL
      ACNT=ACNT+1.                                                   WTVOL
      CALL ORBCAL                                                    WTVOL
C      FIXED THRUST-TO-WEIGHT                                        WTVOL
      ISP(3) = TO27(3)                                               WTVOL
      JUMP = 0                                                       WTVOL
      CALL WTSCH                                                     WTVOL
      CALL SOLVE                                                     WTVOL
      VBODY1 = VBODY                                                 WTVOL
      WGROSO=WGROSS                                                  WTVOL
      WFUOXO=WFUOX                                                   WTVOL
      OTTOT=TTOT                                                     WTVOL
      OSPLAN = SPLAN                                                 WTVOL
      GO TO 291                                                      WTVOL
  195 CONTINUE                                                       WTVOL
      CALL BODCAL                                                    WTVOL
C                                                                    WTVOL
C     CHECK CONSTANT (T/W)L.O. FLAG                                  WTVOL
C                                                                    WTVOL
      IF( SE(3).EQ.0.) GO TO 200                                     WTVOL
C                                                                    WTVOL
C      FIXED THRUST-TO-WEIGHT                                        WTVOL
  197 SCB(129) = 0.                                                  WTVOL
      C(129) = 0.                                                    WTVOL
      CTHRST = CTHRST * SE(1)/ TB27(2)                               WTVOL
      TB18 = CTHRST                                                  WTVOL
      GO TO 210                                                      WTVOL
C      FIXED THRUST                                                  WTVOL
  200 CTHRST = 0.                                                    WTVOL
      TB18 = 0.                                                      WTVOL
      TBTO = SW(6) * TB36/TO36                                       WTVOL
      SCB(129) = OTTOT*TBTO/TB36                                     WTVOL
      C(129) = SCB(129)                                              WTVOL
  210 ISP(3) = TB27(3)                                               WTVOL
      JUMP = 1                                                       WTVOL
      CALL WTSCH                                                     WTVOL
      CALL SOLVE                                                     WTVOL
      BTTOT = TTOT                                                   WTVOL
      VBODY2 = VBODY                                                 WTVOL
  269 TB51= VBODY2                                                   WTVOL
      TO51 = VBODY1                                                  WTVOL
      TB56 = WGROSS                                                  WTVOL
```

```
      WBOO = WGROSS-T056                                              UH
      T056 = WGROSO                                                   WTVOL
  291 CONTINUE                                                        WTVOL
      IF(ILAST.NE.1  .AND. SW(16) . EQ. 0.) GO TO 292                 WTVOL
      CALL PRINTW                                                     CKOUT
      CALL PRINTV                                                     CKOUT
      CALL PRWTSM                                                     CKOUT
  292 CONTINUE                                                        WTVOL
      IF(JUMP.EQ.1) GO TO 300                                         WTVOL
      CALL ORBSUM                                                     WTVOL
      CALL ORBSTO                                                     WTVOL
      GO TO 195                                                       WTVOL
  300 CONTINUE                                                        WTVOL
      C(3) = SCB(3)                                                   WTVOL
      C(5) = SCB(5)                                                   WTVOL
      C(7) = SCB(7)                                                   WTVOL
      C(146) = SCB(146)                                               WTVOL
      C(31) = SCB(31)                                                 WTVOL
      C(56) = SCB(56)                                                 WTVOL
      C(129) = SCB(129)                                               WTVOL
      C(211)=SCB(211)                                                 WTVOL
      C(67) = SCB(67)                                                 WTVOL
C                                                                     WTVOL
C          CHECK CONST. T/W-LO FLAG                                   WTVOL
C                                                                     WTVOL
      IF( SE(3).EQ.0.) GO TO 400                                      WTVOL
      CTHRST = CTHRST * TB27(2)/ SE(1)                                WTVOL
  400 CALL BOOSUM                                                     WTVOL
      CALL BOOSTO                                                     WTVOL
      CALL TAMPER(1)                                                  WTVOL
      TWO = OTTOT/T056                                                WTVOL
      WRITE (6,1000) T034(3),T056,OTTOT,TWO,TB34(3),TB56,QP(1),SV(13) WTVOL
      IF(SE(3) EQ.1.) GO TO 500                                       WTVOL
      IF(ACNT.GE.SW(19)) GO TO 500                                    WTVOL
C  ***  THIS LOOP ADJUSTS CTHRST (ORBITER T/W) TO OBTAIN              WTVOL
C       SPECIFIED LIFTOFF T/W  ***                                    WTVOL
      IF(SW(2).EQ.1.) BTWTOL=SW(18)                                   WTVOL
      IF( ABS(SV(13)-SW(17)) .LE. BTWTOL ) GO TO 500                  WTVOL
      T018 = T018/SV(13) * SW(17)                                     WTVOL
      GO TO 101                                                       WTVOL
  500 CONTINUE                                                        WTVOL
      ACNT=0.                                                         WTVOL
      ACEL=ACEL+1.                                                    WTVOL
      IF(GWREQ.LE.0.) GO TO 800                                       WTVOL
C  ***  THIS LOOP ADJUSTS PAYLOAD WEIGHT TO OBTAIN                    WTVOL
C       SPECIFIED GROSS LIFTOFF WEIGHT  ***                           WTVOL
      DIFF=GWREQ-TB56                                                 WTVOL
      TOM=DIFF/GWREQ                                                  WTVOL
      IF(ABS(TOM) .LT.BIB ) GO TO 504                                 WTVOL
      IF(ACEL.GT.1.) GO TO 502                                        WTVOL
      SLOPE=-21. 1                                                    WTVOL
      IF(SW(3) .GT.1. .OR. GUS.GT.0.) SLOPE=SW(8)                     WTVOL
      GO TO 505                                                       WTVOL
  502 SLOPE = (DIFF - SDIFF)/(SCO(103) - SAPAV)                       WTVOL
  505 CONTINUE                                                        WTVOL
      WRITE(6,1001) SCO(103),DIFF,SLOPE                               WTVOL
      SW(8) = SLOPE                                                   WTVOL
      SDIFF = DIFF                                                    WTVOL
      SAPAV = SCO(103)                                                WTVOL
      SCO(103) = SCO(103) - DIFF/SLOPE                                WTVOL
      IF(ACEL.GT.6.) GO TO 503                                        WTVOL
      GO TO 101                                                       WTVOL
  503 WRITE(6,1002)                                                   WTVOL
      GO TO 600                                                       WTVOL
  504 WRITE(6,1001) SCO(103),DIFF,SW(8)                               WTVOL
      TOM = DIFF/GWREQ                                                WTVOL
      WRITE(6,1003) TOM                                               WTVOL
      GO TO 600                                                       WTVOL
  800 CONTINUE                                                        WTVOL
      IF(SE(3).EQ.1 .OR.SW(19).GT.0.) GO TO 600                       WTVOL
```

```
      C     *****THIS LOOP ADJUSTS ORBITER MR(3) TO COMPENSATE FOR CHANGE IN T/W   WTVOL
      C          AT LIFTOFF.          *****                                       WTVOL
            SQ(15,1)=SQ(15,2)                                                     WTVOL
            SQ(15,2)=SV(13)-1.                                                    WTVOL
      C       SV(13) IS LIFTOFF THRUST-TO-WEIGHT                                  WTVOL
            IF(ACEL.GE.2.) GO TO 600                                              WTVOL
            IF(ACEL.EQ.1..AND.SW(3).LE.1..AND.GUS.EQ.0.) GO TO 600                WTVOL
            DIFF= SQ(15,2)-SQ(15,1)                                               WTVOL
            IF(ABS(DIFF).LT.BIB) GO TO 600                                        WTVOL
            SLOPE = 500./SE(7)/32.174                                             WTVOL
            X=TO34(3)*SLOPE/SQ(15,2) - .2                                         WTVOL
      C       THE CONSTANT -.2 REFLECTS WEIGHT REDUCTION DUE TO QMAX DECREASE     WTVOL
      C     *****  X IS AN ESTIMATE OF THE FEEDBACK OF T/W VARIATIONS             WTVOL
      C          THROUGH CHANGES IN MUO AND QMAX.  *****                          WTVOL
            IF(X.GE..88) GO TO 803                                                WTVOL
            IF(X.GT..44)  X=.44                                                   WTVOL
            DTWT=-DIFF/SQ(15,1)/SQ(15,2)/(1.-X)                                   WTVOL
            DTWT = DIFF/ (1.-X)                                                   WTVOL
            RATIO = 1. + DTWT / SQ(15,2)                                          WTVOL
            OLD=TO34(3)                                                           WTVOL
            TO34(3) = OLD * RATIO**(-2.*SLOPE)                                    WTVOL
            DMU= TO34(3)-OLD                                                      WTVOL
            WRITE(6,1004) DMU,X,DTWT                                              WTVOL
      C     *****  ADJUST THE QMAX ESTIMATE  *****                                WTVOL
            SE(6) = SE(6) + SE(4)* DTWT                                           WTVOL
            QMX =  SE(6)                                                          WTVOL
            GO TO 101                                                             WTVOL
        803 WRITE(6,1008)                                                         WTVOL
            EROR = 1.1                                                            WTVOL
       1008 FORMAT(5X,50H ******* RUN ABORTED IN WTVOL, FIXED THRUST LOW    )     WTVOL
            GO TO 900                                                             WTVOL
      C                                                                           WTVOL
        600 CONTINUE                                                              WTVOL
            IF(ACEL.GT.6.) GO TO 900                                              WTVOL
            ACEL = 0.                                                             WTVOL
            IF(WOREQ.LE.0.) GO TO 700                                             WTVOL
      C     ***  THIS LOOP ADJUSTS BOOSTER MASS RATIO TO OBTAIN                   WTVOL
      C          SPECIFIED ORBITER GROSS WEIGHT.  ***                             WTVOL
            DW=WOREQ-TO56                                                         WTVOL
            ERR=DW/WOREQ                                                          WTVOL
            IF(ABS(ERR) .LT. BOB) GO TO 604                                       WTVOL
            IF(GUS.GT.0.) GO TO 602                                               WTVOL
            S= .54 *TO56 * TO34(3)/TB34(3)                                        WTVOL
            IF(SW(3).GT.1.) S = SQ(11,1)                                          WTVOL
            GUS=0.                                                                WTVOL
            GO TO 605                                                             WTVOL
        602 S = (D W-SD W)/(TB34(3)-SMR)                                          WTVOL
        605 WRITE(6,1005) D W,S                                                   WTVOL
            SQ(11,1) = S                                                          WTVOL
            SD W = D W                                                            WTVOL
            SMR = TB34(3)                                                         WTVOL
            TB34(3) = TB34(3) - D W/S                                             WTVOL
            RPAR=SQ(19,3)                                                         WTVOL
            IF(RPAR.GT.100.)SCB(214)=SCB(214)+SE(5)*(TB34(3)-SMR)/RPAR            WTVOL
            TO34(3)= TO34(3)*SMR / TB34(3)                                        WTVOL
            IF(GUS.GT.6.) GO TO 603                                               WTVOL
            GUS = GUS + 1.                                                        WTVOL
            GO TO 101                                                             WTVOL
        603 WRITE(6,1002)                                                         WTVOL
        604 WRITE(6,1005) DW    ,SQ(11,1)                                         WTVOL
            GO TO 900                                                             WTVOL
        700 IF(WPOREQ.LE.0.) GO TO 900                                            WTVOL
      C     ***  THIS LOOP ADJUSTS BOOSTER MASS RATIO TO OBTAIN                   WTVOL
      C          SPECIFIED ORBITER PROPELLANT WEIGHT                              WTVOL
      C          INCLUDES FPR, EXCLUDES ON-ORBIT DELTA V   ***                    WTVOL
            DW=WPOREQ-WFUOXO                                                      WTVOL
            ERR=DW/WPOREQ                                                         WTVOL
            IF(ABS(ERR) .LE. BOB) GO TO 704                                       WTVOL
            IF(GUS.GT.0.) GO TO 702                                               WTVOL
            S=  .690*WFUOXO*TO34(3) / TB34(3)                                     WTVOL
            IF(SW(3).GT.1.) S = SQ(11,1)                                          WTVOL
            GUS=0.                                                                WTVOL
```

```
      GO TO 705                                                         WTVOL
  702 S = (DW  -SDW  )/(TB34(3)-SMR)                                    WTVOL
  705 WRITE(6,1006) DW  ,S                                              WTVOL
      SQ(11,1) = S                                                      WTVOL
      SD W = D W                                                        WTVOL
      SMR = TB34(3)                                                     WTVOL
      TB34(3) = TB34(3) - DW  /S                                        WTVOL
      RPAR=SQ(19,3)                                                     WTVOL
      IF(RPAR.GT.100.)SCB(214)=SCB(214)+SE(5)*(TB34(3)-SMR)/RPAR        WTVOL
      TO34(3)= TO34(3)*SMR / TB34(3)                                    WTVOL
      IF(GUS.GT.6.) GO TO 703                                           WTVOL
      GUS = GUS + 1.                                                    WTVOL
      GO TO 101                                                         WTVOL
  703 WRITE(6,1002)                                                     WTVOL
  704 WRITE(6,1006) DW,SQ(11,1)                                         WTVOL
  900 CONTINUE                                                          WTVOL
      IF(SW(2).EQ.1.) ILAST=1                                           WTVOL
      IF(ILAST.EQ.1) GO TO 101                                          WTVOL
  950 CONTINUE                                                          WTVOL
      CALL TAMPER(2)                                                    WTVOL
      RETURN                                                            WTVOL
 1000 FORMAT (15HOORBITER  MU = F8.5,13H    WEIGHT = F9.0,13H    THRUST WTVOL
     1= F9.0,10H    T/W = F9.6/                                         WTVOL
     2          15H BOOSTER  MU = F8.5,13H    WEIGHT = F9.0,13H    THRUST WTVOL
     3= F9.0,10H    T/W = F9.6)                                         WTVOL
 1001 FORMAT(5X,11HPAYLOAD =  F10.2,5X,14HLO WT DIFF =  F12.2,5X,10HSLOP WTVOL
     1E  =  F12.4///)                                                   WTVOL
 1002 FORMAT(5X,36H***** ITERATION COUNT EXCEEDED *****/)               WTVOL
 1003 FORMAT(5X,15HCONVERG ERR =  F10.6///)                             WTVOL
 1004 FORMAT( 8X,5HOMU = F8.5, 5H  X = F8.4, 8H  DTWT = F9.6 / )        WTVOL
 1005 FORMAT(5X,16HORB  WT DIFF =  F10.1,5X,9HSLOPE =  F10.1//)         WTVOL
 1006 FORMAT(5X,25HORB PROPELLANT WT DIFF = F10.1,5X,8HSLOPE = F10.1//) WTVOL
      END                                                               WTVOL
```

Hunting Procedure (PØWELL) 10

1616

Section 1
POWELLS METHOD

The purpose of Subroutine POWELL is to find the values of N parameters, given by the vector—say,

$$\overline{Z} = (z_1, z_2, \ldots, z_N) \qquad (1.1)$$

so that the value of a function of these parameters—say, $F(\overline{Z})$—is a minimum subject to constraints on the parameters of the form,

$$e_i \le z_i \le h_i\ , \ i=1,\ 2,\ \ldots,\ N \qquad (1.2)$$

The inequality constraints on the parameters (Equation 1.1) are handled by the BOX TRANSFORMATION where a change of variable is made from the z_i to a new set of parameters—say,

$$\overline{X} = (x_1, x_2, \ldots, x_N)$$

by the formula,

$$x_i = \text{ARCSIN} \left\{\sqrt{(z_i - e_i)/(h_i - e_i)}\right\} \qquad (1.3)$$

Subroutine POWELL uses the x_i as a new set of independent parameters which have no inequality constraints imposed upon them. In terms of $\overline{X}$-space, the algorithm minimizes a function $f(\overline{X})$ related to $F(\overline{Z})$ by the equation:

$$f(\overline{X}) = F(\overline{Z}(\overline{X}))$$

1617

where $Z(X)$ is defined by the inverse transformation,

$$z_i = e_i + (h_i - e_i) \sin^2 x_i \qquad (1.4)$$

$$i = 1, 2, \ldots, N$$

1.1 BASIC ALGORITHM

The basic procedure is as follows (Reference 2):

1. Choose a best known approximation to the minimum of f—say, $\overline{X}_o$.

   Let $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N$ be unit vectors in the coordinate directions; then for $r = 1, 2, \ldots, N$ calculate λ_r so that $f(\overline{X}_r)$ is a minimum where $\overline{X}_r$ is defined by

   $$\overline{X}_r = \overline{X}_{r-1} + \lambda_r \overline{\varepsilon}_r \qquad (1.5)$$

2. When $\overline{X}_N$ is determined, find a λ_c so that $f(\overline{X}_N + \lambda_c \overline{\varepsilon}_{N+1})$ is a minimum, where $\overline{\varepsilon}_{N+1}$ is the unit vector in the combined direction:

   $$(\overline{X}_N - \overline{X}_o).$$

3. Rotate the N+1 vectors, $\overline{\varepsilon}_1, \overline{\varepsilon}_2, \ldots, \overline{\varepsilon}_N, \overline{\varepsilon}_{N+1}$, by replacing $\overline{\varepsilon}_r$ by $\overline{\varepsilon}_{r+1}$ for $r = 1, 2, 3, \ldots, N$.

4. Replace $\overline{X}_o$ by $\overline{\varepsilon}_{N+1} + \lambda_c \overline{\varepsilon}_{N+1}$ and return to Step 1.

The sequence of Steps 1 through 4 will be called a <u>major iteration.</u>

As indicated by the above steps, the procedure requires N+1 minimizations per major iteration.

1.2 THE MINIMIZATION ALGORITHM

The minimization algorithm employed in Subroutine POWELL uses an accelerated linear search until a set of three points which define a concave upward curve is achieved. It then uses these three points to initiate a quadratic search for the minimum.

1618

In general, assume we are given a point $\overline{X}_o$ in X-space and a unit vector $\bar{\varepsilon}$ along which the minimization is to take place;

$$\bar{\varepsilon} = (u_1, u_2, \ldots, u_N) \text{ with } |\bar{\epsilon}| = 1$$

Associated with each independent parameter, x_i, is a nominal step size— say, s_i.

Now, define $k_{\bar{\varepsilon}}$ by

$$k_{\bar{\varepsilon}} = \left[\sum_{i=1}^{N} \left(\frac{u_i}{s_i} \right)^2 \right]^{-\frac{1}{2}} \qquad (1.6)$$

Now the step taken along $\bar{\varepsilon}$ on the accelerated linear search portion of the minimization algorithm is a function of:

1. $p = \{$number of previous steps taken along $\bar{\varepsilon}.\}$ (including the null step).
2. $k_{\bar{\varepsilon}}$ as defined above.
3. $q = \{$number of the current major iteration$\}$.

Now, calling the current step $S(p, k_{\bar{\varepsilon}}, q)$, we have

$$S(p, k_{\bar{\varepsilon}}, q) = 2^{p-1} (k_{\bar{\varepsilon}})/q \qquad (1.7)$$

(See Appendix A)

The process of stepping along $\bar{\varepsilon}$ is from the current initial point $\overline{X}_o$; each new step being twice the previous step.

Now, if the accelerated linear search is in a descent direction, then the search continues until a point $\overline{X}_p$ is found so that

$$\overline{X}_p > X_{p-1} \ (p>1) \qquad (1.8)$$

The minimum along $\bar{\varepsilon}$ (for a true quadratic) is given by $(\bar{X}_m, f(\bar{X}_m))$ where

$$\bar{X}_m = \bar{X}_o + s_m \bar{\varepsilon} \tag{1.12}$$

Since in general f is not quadratic, the quadratic search procedure is iterative and continues as follows. (Assume $s_i < s_j < s_k$.)

1. If $s_m > s_k$ or $s_m < s_i$, throw the worst point of Equation (1.10)—the point with highest payoff—and replace that point by $(\bar{X}_m, f(\bar{X}_m))$ for the next quadratic fit.
2. If $s_i \leq s_m \leq s_k$, we have two subcases to consider:
   A. If the worst point is $(\bar{X}_k, f(\bar{X}_k))$
      1) If $s_m > s_j$, then throw away $(\bar{X}_i, f(\bar{X}_i))$ and replace it with $(\bar{X}_m, f(\bar{X}_m))$.
      2) If $s_m < s_j$, then throw away the worst point and replace it by $(\bar{X}_m, f(\bar{X}_m))$.
   B. If the worst point is $(\bar{X}_i, f(\bar{X}_i))$
      1) If $s_m > s_j$, then throw away the worst point and replace it with $(\bar{X}_m, f(\bar{X}_m))$.
      2) If $s_m < s_j$, then throw away $\bar{X}_k$ and replace it with $(\bar{X}_m, f(\bar{X}_m))$.

1.3 CONVERGENCE AND STOPPING CRITERIA

Convergence and stopping are controlled by the following conventions.

1. Maximum number of major iterations allowed is 20.
2. Maximum number of quadratic fits allowed in any one search is 7.
3. A quadratic search is called converged if
   A. The current payoff value is less than or equal to the previously computed payoff value.
   B. The difference between the previous payoff and the current payoff is less than 20 percent of the previous payoff (5 percent if there is only one independent parameter).
   C. After A and B are satisfied, the algorithm tries twice more to get the current payoff less than the value obtained at the start of the current major iteration. If it fails, then only A and B are required for convergence. (Once convergence is achieved,

1620

where

$$\overline{X}_p = \overline{X}_o + S(p, k_{\overline{\varepsilon}}, q) \tag{1.9}$$

Under these circumstances, the last three points, $\overline{X}_p$, $\overline{X}_{p-1}$, and $\overline{X}_{p-2}$, are fitted with a quadratic for a minimum. (This begins the quadratic search procedure for this case.)

If, however, the accelerated linear search is in an ascent direction—i.e., $\overline{X}_1 > \overline{X}_o$—then the search continues for $\overline{X}_2$ in the direction $-\overline{\varepsilon}$ by the magnitude of the step which produced $\overline{X}_1$. If the triple $\overline{X}_1$, $\overline{X}_2$, $\overline{X}_3$ yields a concave upward set, then these points are fitted with a quadratic for a minimum.

If the triple $\overline{X}_1$, $\overline{X}_2$, $\overline{X}_3$ does not yield a concave upward set, then the descent portion of the algorithm is used, but now in the opposite direction. Once three satisfactory points are determined, the procedure goes into a quadratic search mode. This will now be considered.

Assume that somewhere along the line the accelerated linear search yields a concave upward set of three points—say,

$$(\overline{X}_i, f(\overline{X}_i)), \ (\overline{X}_j, f(\overline{X}_j)), \ (\overline{X}_k, f(\overline{X}_k)) \tag{1.10}$$

with associated steps s_i, s_j, s_k, a quadratic fit is now made using the formula for the step to the minimum, s_m, given by

$$s_m = \frac{1}{2}\left\{\frac{A_1 + A_2 + A_3}{B_1 + B_2 + B_3}\right\} \tag{1.11}$$

where

$$A_1 = \left(\overline{X}_j^2 - \overline{X}_k^2\right) f(\overline{X}_i) \qquad B_1 = (\overline{X}_j - \overline{X}_k) f(\overline{X}_i)$$

$$A_2 = \left(\overline{X}_k^2 - \overline{X}_i^2\right) f(\overline{X}_j) \qquad B_2 = (\overline{X}_k - \overline{X}_i) f(\overline{X}_j)$$

$$A_3 = \left(\overline{X}_i^2 - \overline{X}_j^2\right) f(\overline{X}_k) \qquad B_3 = (\overline{X}_i - \overline{X}_j) f(\overline{X}_k)$$

the best of the points generated is used to continue the algorithm.)

4. If $\underline{N > 1}$, convergence of the total problem is achieved if the changes in the independent parameters are small (determined by an input tolerance).

5. If $\underline{N = 1}$, convergence of the quadratic search is defined as convergence of the total problem.

1622

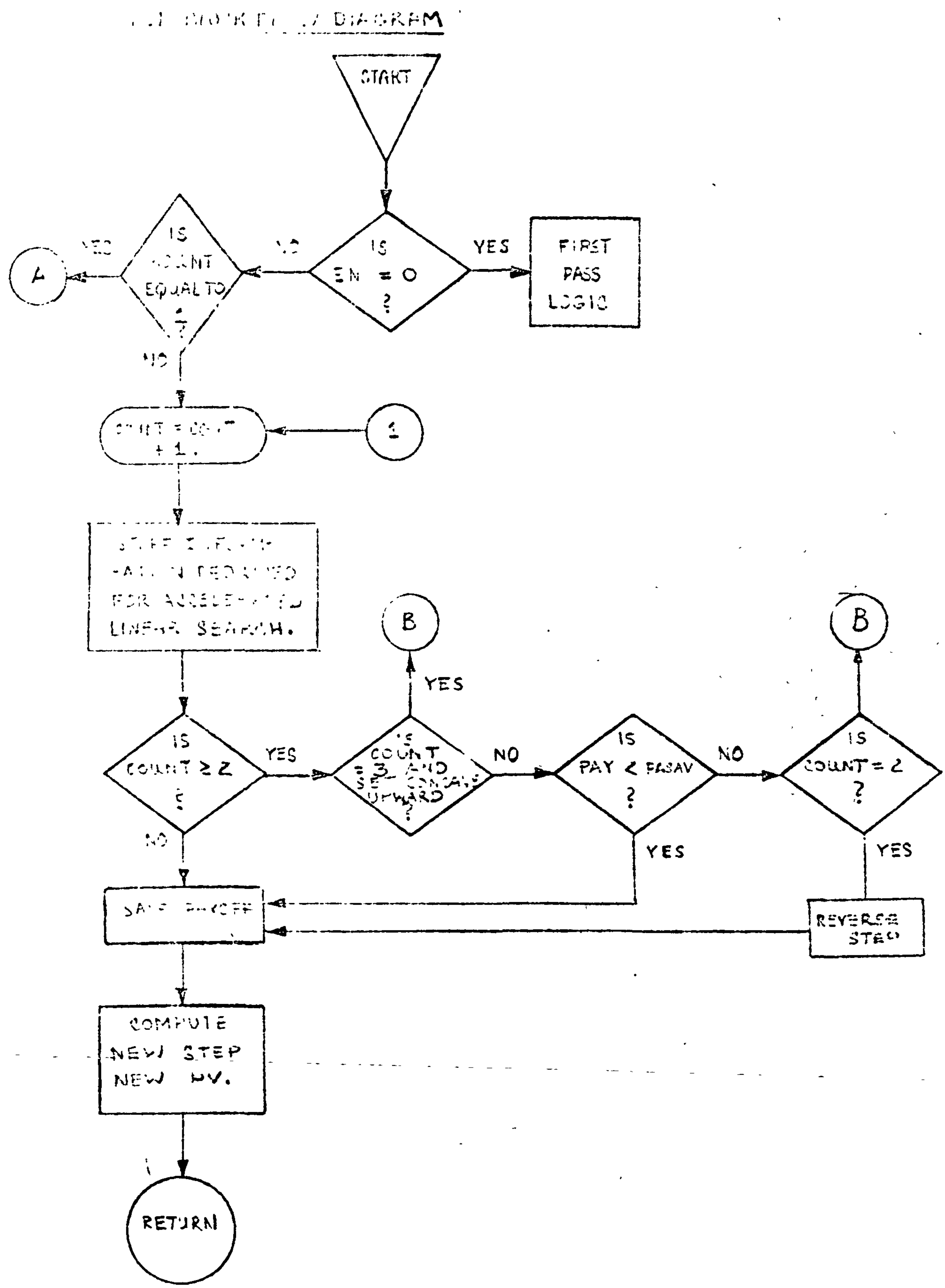
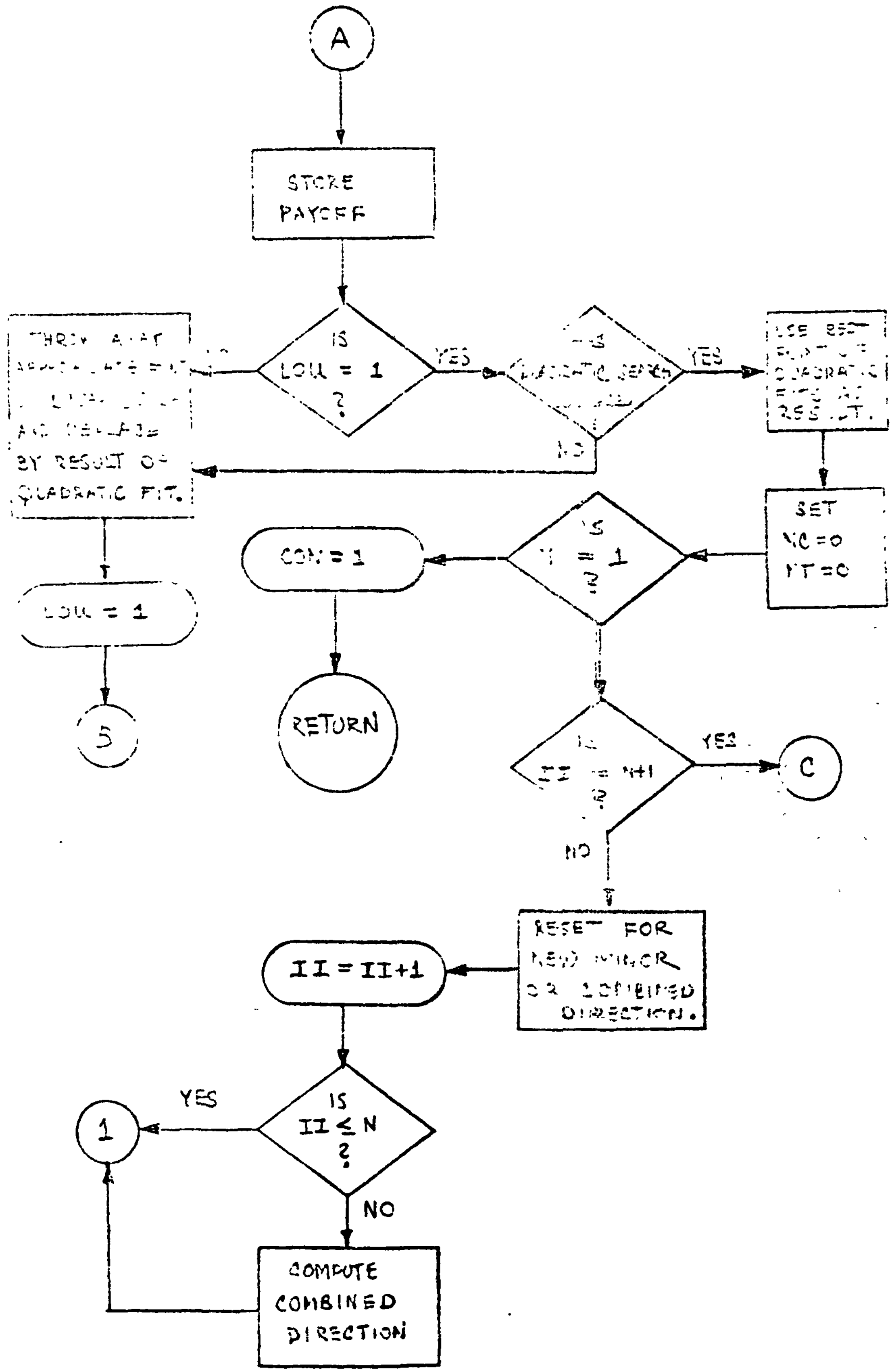
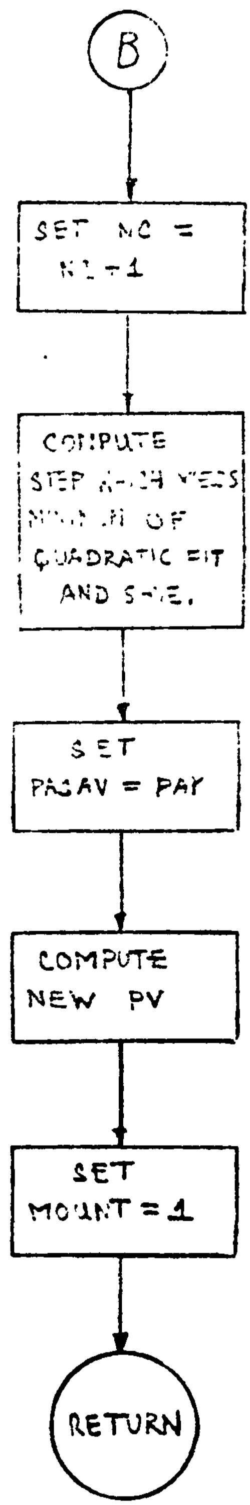
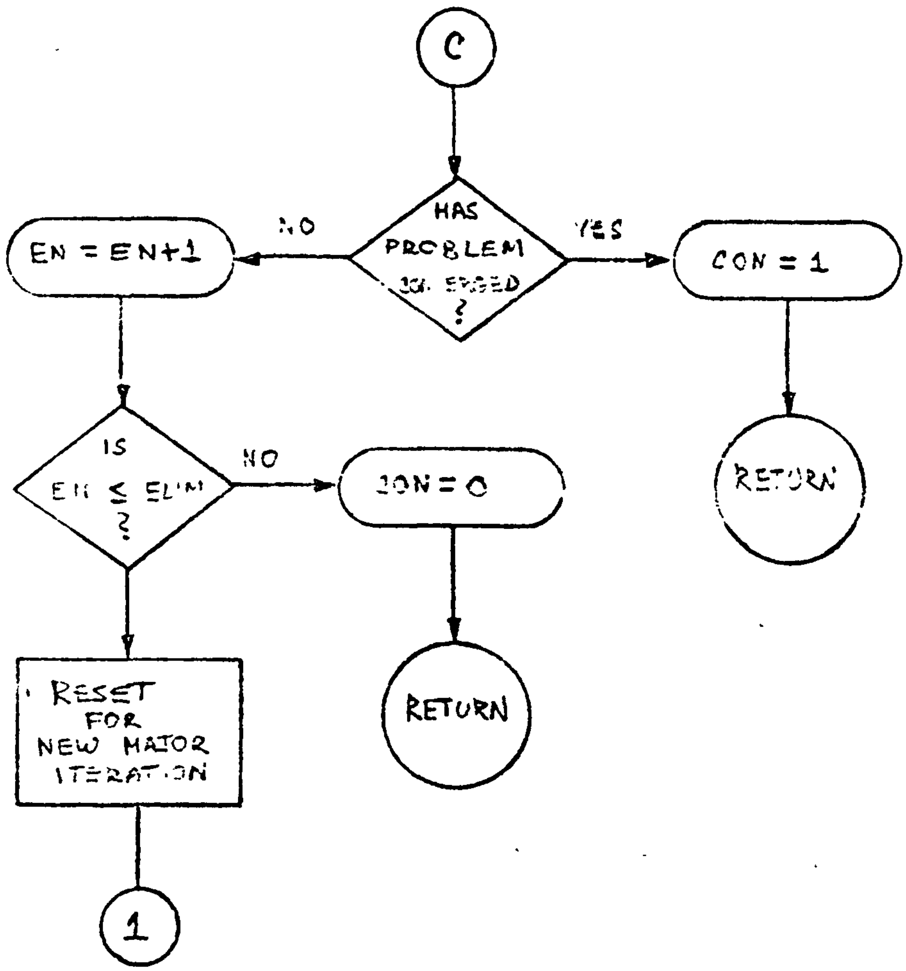


DIAGRAM
START
IS
EN = 0
?
YES
FIRST
PASS
LOGIC
IS
EQUAL TO
A
NO
+ 1.
LINEAR SEARCH.
B
YES
IS
COUNT ≥ 2
NO
YES
COUNT
= 3 AND
UPWARD
NO
IS
PAY < PASAV
?
NO
IS
COUNT = 2
?
YES
YES
REVERSE
STEP
COMPUTE
NEW STEP
NEW PV.
RETURN


A
STORE PAYOFF
IS LOW = 1 ?
YES
YES
BY RESULT OF QUADRATIC FIT.
SET NC=0
LOW = 1
CON = 1
5
RETURN
YES
C
NO
RESET FOR NEW MINOR OR COMBINED DIRECTION.
II = II+1
YES
IS II ≤ N ?
1
NO
COMPUTE COMBINED DIRECTION


B
SET NC =
COMPUTE
QUADRATIC FIT
SET
PASAV = PAY
COMPUTE
NEW PV
SET
RETURN

1625


C
HAS PROBLEM CONVERGED ?
NO
YES
EN = EN+1
CON = 1
IS EN ≤ ELIM ?
NO
CON = 0
RETURN
RESET FOR NEW MAJOR ITERATION
RETURN
1

3.0 LISTING

```
              SUBROUTINE POWELL(PV,N,STEP,EPP,PAY)
000010        COMMON /POWEL/
             *DEL (11) ,II         ,COUNT     ,XT(3)    ,AT (3)     ,PVS(10)   ,
             *PASAV    ,CON    ,    LOU    ,EN     ,PPN(10)    ,
             *SC     ,MOUNT    ,GG(11,10)
             *,PAO,NT,NC,DD,D(7),V(7)
000010        COMMON/SIZING/DUMMY(320),ELIM
000010        DIMENSION PV(1),STEP(1)
             *,EPP(1)
000010        INTEGER  COUNT,CON,EN
000010        INTEGER ELIM
000010        IF(EN,GE,1) GO TO 19
         C
         C             FIRST PASS LOGIC
         C
000012        NC=0
000013        NT=0
000014        PAO = PAY
000015        SC=1.
000016        CON = -1
000017        LOU=0
000020        COUNT =0
000021        MOUNT =0
000022        II=1
000023        EN=1
000024        DO 5  I=1,3
000025        XT(I) = 0.
000026      5 AT(I) = 0.
000031        DO 6  I=1,11
000032      6 DEL(I) = 0.
000035        DO 10  I=1,N
000036        PVS(I) =PV(I)
000040     10 PPN(I) = PV(I)
000044        NP1 =N+1
000046        DO 17 I=1,NP1
000047        DO 17  J =1,N
000050        IF(I.EQ.J)  GO TO 15
000051        GG(I,J)= 0.
000054        GO TO 17
000054     15 GG(I,J)= 1.
000060     17 CONTINUE
         C
         C             CHECK VALUE OF MOUNT.
         C                          1.) IF MOUNT EQUAL ZERO STORE INFORMATION
         C                              REQUIRED FOR ACCELERATED LINEAR SEARCH.
         C                          2.) IF MOUNT EQUAL ONE CONTINUE WITH QUADRATIC
         C                              SEARCH.
         C
000065     19 IF(MOUNT,EQ.1) GO TO 100
000067     20 COUNT = COUNT +1
000071        XT(1) = XT(2)
000072        XT(2) = XT(3)
000073        XT(3) = DEL(II)
000075        AT(1) = AT(2)
000076        AT(2) = AT(3)
000077        AT(3) = PAY
000100        IF(COUNT,GE.2) GO TO 70
```

1627

```
          C
          C               SAVE PAYOFF TO CHECK IF ACCELERATED LINEAR SEARCH
          C               SHOULD BE ENDED.
          C
 000103       52 PASAV = PAY
          C
          C               COMPUTE STEP SIZE FOR USE IN ACCELERATED LINEAR SEARCH,
          C               AND COMPUTE NEW VALUE OF PV.
          C
 000104          VAL = 0.
 000105          DO 53  I=1,N
 000107       53 VAL = VAL + (GG(II,I)/STEP(I))**2
 000123          DEL(II) =1./SQRT(VAL)
 000130          DEL(II) = 2.**COUNT*DEL(II)/(2.*EN)
 000137          DEL(II) = SC * DEL(II)
 000142          DO 55 I =1,N
 000146       55 PV(I)= PPN(I) + DEL(II)*GG(II,I)
 000166          RETURN
          C
          C               COMPARE VALUES OF PAY A PASAV.
          C                         1.) IF PAY IS LESS THAN PASAV, SET PASAV EQUAL
          C                             TO PAY, COMPUTE NEW STEP SIZE AND NEW VALUE
          C                             OF PV.
          C                         2.) IF PAY IS GREATER THAN OR EQUAL TO PASAV,
          C                                   A.)  IF COUNT IS EQUAL TO TWO, SET
          C                                        PASAV EQUAL TO PAY, REVERSE THE
          C                                        DIRECTION OF NEW STEP,COMPUTE
          C                                        NEW PV.
          C                                   B.)  IF COUNT NOT EQUAL TO TWO BEGIN
          C                                        QUADRATIC SEARCH.
          C
 000167       70 IF(PAY.GT.AT(1).AND.SS.EQ.-.5.AND.COUNT.EQ.3) GO TO 75
 000203          IF(PAY.LT.PASAV) GO TO 52
 000205          IF(COUNT.EQ.2) GO TO 90
          C
          C               BEGIN QUADRATIC SEARCH.
          C
 000207       75 NC=NC+1
 000211          AA =(XT(2) - XT(3))*AT(1)
 000214          BB =(XT(3) - XT(1))*AT(2)
 000217          CC =(XT(1) - XT(2))*AT(3)
 000222          AS =(XT(2)**2 -XT(3)**2 )*AT(1)
 000225          BS =(XT(3)**2 -XT(1)**2 )*AT(2)
 000231          CS =(XT(1)**2 -XT(2)**2 )*AT(3)
 000234          DD= (AS+BS+CS)/(AA+BB+CC)
 000242          DD = .5*DD
 000243          D(NC)=DD
 000245          PASAV = PAY
 000246          DO 85 I=1,N
 000250       85 PV(I) = PPN(I) +DD*GG(II,I)
 000256          MOUNT = 1
 000257          RETURN
          C
          C               REVERSE DIRETION OF STEP.
          C
 000270       90 SC = -.5
 000272          GO TO 52
          C
```

```
                C             CHECK VALUE OF LOU,
                C                         1.) IF LOU EQUAL TO ONE, CHECK CONVERGENCE OF
                C                             QUADRATIC SEARCH
                C                         2.) IF LOU EQUAL TO ZERO, THROW AWAY WORST
                C                             VALUE OF ACCELERATED LINEAR SEARCH AND
                C                             REPLACE BY RESULT OF QUADRATIC SEARCH
                C
  000272          100 V(NC)=PAY
  000274              IF(LOU.EQ.1) GO TO 150
  000276          105 PMAX = AMAX1(AT(1), AT(2), AT(3))
  000304              DO 120 I=1,3
  000306              IF(PMAX.EQ.AT(I)) GO TO 121
  000310          120 CONTINUE
  000312          121 JM = I
  000314              PMIN =AMIN1(AT(1),AT(2),AT(3))
  000321              DO 130 I=1,3
  000323              IF(PMIN.EQ.AT(I)) GO TO 131
  000325          130 CONTINUE
  000327          131 JS=I
  000331              JL = MOD(JM+JS,4)
  000335              IF(JL.EQ.0) JL =2
  000337              FM1 =AMAX1(XT(1),XT(2),XT(3))
  000345              FM2 =AMIN1(XT(1),XT(2),XT(3))
  000352              IF(DD.GT.FM1.OR.DD.LT.FM2) GO TO 141
  000353              IF(XT(JM).EQ.FM1.AND.XT(JL).EQ.FM2) GO TO 137
  000374              IF(XT(JM).EQ.FM2.AND.XT(JL).EQ.FM1) GO TO 135
  000405              GO TO 141
  000405          135 IF(DD.GT.XT(JS)) GO TO 141
  000411              GO TO 145
  000411          137 IF(DD.GT.XT(JS)) GO TO 145
  000415          141 XT(JM)=DD
  000417              AT(JM)=PAY
  000421              GO TO 149
  000421          145 XT(JL)=DD
  000423              AT(JL)=PAY
  000425          149 LOU = 1
  000426              GO TO 75
  000427          150 IF(NC.GE.7) GO TO 152
  000432              IF(PAY.GT.PASAV) GO TO 105
  000435              TT =.2
  000436              IF(N.EQ.1) TT =.05
  000441              IF(PASAV-PAY.GT.TT*PASAV) GO TO 105
  000446              IF(NT.EQ.2) GO TO 152
  000450              NT =NT+1
  000451              IF(PAY.GT.PA0) GO TO 105
                C
                C             IF N EQUAL TO ONE, SET CON EQUAL TO ONE AND RETURN
                C
  000455          152 IM=1
  000456              VMIN=V(1)
  000460              DO 153  I=2,NC
  000461              IF(VMIN.GT.V(I)) IM=I
  000465          153 VMIN =V(IM)
  000472              DD = D(IM)
  000474              DO 154  I=1,N
  000475          154 PV(I) =PPN(I)+ DD * G3(II,I)
  000513              PAY = V(IM)
  000515              NT=0
```

1629

```
000516          NC=0
000517          IF(N.EQ.1) GO TO 252
000521          IF(II.EQ.N+1) GO TO 200
          C
          C               RESET FOR NEW MINOR OR COMBINED SEARCH.
          C
000523          DO 155  I=1,N
000524      155 PPN(I) = PPN(I) + DD*GG(II,I)
000535          SC=1.
000536          COUNT =0
000537          MOUNT = 0
000540          LOU = 0
000541          NP1=N+1
000543          DO 160 I=1,NP1
000544      160 DEL(I) = 0.
000547          II=II+1
000551          IF(II.LE.N) GO TO 20
          C
          C               COMPUTE COMBINED DIRETION.
          C
000553          GVAL=0
000554          DO 175 I = 1,N
000555      175 GVAL = GVAL+(PPN(I)-PVS(I))**2
000563          GVAL = SQRT(GVAL)
000565          DO 180 I = 1,N
000571      180 GG(II,I) = (PPN(I)-PVS(I))/GVAL
000610          GO TO 20
          C
          C               CHECK CONVERGENCE  OF PROBLEM.
          C
000611      200 DO 250 I =1,N
000613      250 IF(ABS(PVS(I)-PV(I)).GT.EPP(I)) GO TO 255
000624      252 CON = 1
000625          RETURN
          C
          C               INCREMENT EN.
          C                       1.) IF EN IS LESS THAN OR EQUAL TO ELIM RESET
          C                           FOR NEW MAJOR
          C                        2.) IF NOT,SET CON EQUAL TO ZERO AND RETURN.
          C
000626      255 EN=EN+1
000630          IF(EN.LE.ELIM ) GO TO 260
000632          CON = 0
000633          RETURN
000634      260 DO 261 I = 1,N
000636          PVS(I) =PV(I)
000640      261 PPN(I) = PV(I)
000644          PAO = PAY
000645          II =1
000646          SC = 1.
000647          COUNT=0
000650          MOUNT = 0
000651          LOU   = 0
000652          NP1=N+1
000654          DO 265  I=1,NP1
000655      265 DEL(I) = 0.
000660          DO 270 I = 1,N
000662          DO 270  J =1,N
```

```
000653      270 GG(I,J)  =G(I+1,J)
000677          GO TO 20
000677          END
```

Section 4
GLOSSARY OF TERMS

Name	Definition
AA	Working variable used to compute minimum of quadratic fit.
AS	Working variable used to compute minimum of quadratic fit.
AT	Array of Dimension 3 which stores the payoff valves of the accelerated linear search used in the quadratic search.
BB	Working variable used to compute minimum of quadratic fit.
BS	Working variable used to compute minimum of quadratic fit.
CC	Working variable used to compute minimum of quadratic fit.
CON	Flag which indicates convergence of problem. 1. If CON = 1, problem has converged 2. If CON = 0, iteration limit exceeded without convergence.
COUNT	Number of steps taken in accelerated linear search (including the null step).
CS	Working variable used to compute minimum of quadratic fit.
D	Array of Dimension 7 which stores the step sizes to a minimum for any single quadratic search.
DD	Step size to a minimum obtained from quadratic search.
DEL	Step used in accelerated linear search.
ELIM	Maximum number of major iterations allowed.
EN	Major iteration counter.
EPP	Convergence tolerance: for N > 1. If the difference between the independent parameters in two successive major iterations are all less than EPP, then the problem has converged.
FM1	Maximum of the three steps used in any one pass of the quadratic search.

1632

Name	Definition
FM2	Minimum of the three steps used in any one pass of the quadratic search.
GG	Array of dimension (10 x 12) which contains the direction vectors.
GVAL	Magnitude of combined direction vector.
II	Counts the number of search directions per major interaction.
JL	Index of the step sizes for the maximum payoff of the three payoff values used in any single pass of the quadratic search.
JM	Index of the step size for the middle valued payoff of the three payoff values used in any single pass of the quadratic search.
JS	Index of the step size for the smallest payoff of the three payoff values used in any single pass of the quadratic search.
MOUNT	Flag so that, 1. If MOUNT equals zero, store information required for accelerated linear search. 2. If MOUNT equals 1, continue with the quadratic search.
N	Number of independent parameters.
NC	Counts number of passes in any single quadratic search (Max = 7).
NT	Counter which limits to two the number of attempts to get payoff less than the value achieved at the start of the current major iteration (after certain weaker convergence criteria have been met).
PAO	Payoff value at start of a major iteration.
PASAV	Previous payoff value.
PAY	Current payoff value.
PMAX	Maximum payoff of the three payoffs used in any single pass of the quadratic search.
PMIN	Minimum payoff of the three payoffs used in any single pass of the quadratic search.
PPN	Vector which defines current search direction.
PV	Independent parameter array (Max = 10).

Name	Definition
PVS	Independent parameter array for previous major iteration.
SC	Multiplier which reverses direction of linear search.
STEP	Nominal step size array (Max = 10).
XT	Array of Dimension 3 which stores the steps of the accelerated linear search to be used in the quadratic search.
V	Array of Dimension 7 which stores the payoff values for any single quadratic search.
VAL	Working variable used to compute $k_{\bar{\varepsilon}}$.
VMIN	Working variable used to compute index of best point in any single quadratic search.

Appendix A

ON THE CHOOSING OF A STEP SIZE

The method for determining the step size $S(p,\ k_{\bar{\varepsilon}},\ q)$ will now be presented.

The nominal steps s_i associated with each independent parameter determine an ellipse in X-space; viz,

$$\sum_{i=1}^{N} \left(\frac{x_i}{s_i}\right)^2 = 1 \qquad \text{(A-1)}$$

Given a direction determined by a unit vector,

$$\bar{\varepsilon} = (u_1,\ u_2,\ \ldots,\ u_N)$$

we wish to find a constant $k_{\bar{\varepsilon}}$ so that, $k_{\bar{\varepsilon}}\varepsilon$ lies on the ellipse in Equation (A-1).

Therefore the following equation must be satisfied;

$$\sum_{i=1}^{N} \left(\frac{k_\varepsilon u_i}{s_i}\right)^2 = 1 \qquad \text{(A-2}$$

Solving for k_ε we obtain,

$$k_\varepsilon = \left[\sum_{i=1}^{N} \left(\frac{u_i}{s_i}\right)^2\right]^{-1/2} \qquad \text{(A-3)}$$

Therefore, the step taken in the direction $\bar{\varepsilon}$ should have magnitude $k_{\bar{\varepsilon}}$.

To accelerate the linear search, k_ε is multiplied by the factor 2^{p-1}, but, also, to cut down the size of the step as convergence is approached, $2^{p-1}\, k_\varepsilon$ is divided by q (p = number of previous steps; q = major iteration number). Therefore, we have

$$S(p,\ k_{\bar{\varepsilon}},\ q) = 2^{p-1}\, k_{\bar{\varepsilon}}/q \tag{A-4}$$

1636

Appendix B
NUMERICAL RESULTS

The subroutine was tested on Rosenbrocks function of two variables x_1, x_2; viz,

$$f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 \qquad \text{(B-1)}$$

with the starting conditions

$$x_1 = -1.2 \ , \ x_2 = 1 \qquad \text{(B-2)}$$

Two major cases were considered, each with its own spectrum of step sizes (the s_i). All cases converged but with a varying number of function evaluations. (In all cases $s_1 = s_2$.)

The converged values to the problem are

$$x_1 = x_2 = 1 \ , \ f(x_1 x_2) = 0 \ . \qquad \text{(B-3)}$$

The number of function evaluations to get $f(X_1, X_2) < 10^{-5}$ and $f(X_1, X_2) < 10^{-9}$ is tabulated in the following charts.

Case 1: NO CONSTRAINTS ON INPUTS

Step ($s_1 = s_2$)	No. of evaluations for $f < 10^{-5}$	No. of evaluations for $f < 10^{-9}$
.1	226	237
.4	189	197
.5	167	181
.6	188	197

Case 2: INPUT CONSTRAINTS

The constraints are,

$$-5 \le x_i \le 5 \quad , \quad i = 1, 2$$

Step ($s_1 = s_2$)	No. of evaluations for $f < 10^{-5}$	No. of evaluations for $f < 10^{-9}$
.05	167	177
.1	127	155
.2	269	285
.4	253	270

Ground Rules for Modifying PADS 11

REFERENCES

1. M. J. Box. A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems. Computer Journal, Vol. 8, No. 1, 1965, pp. 303-307.

2. M. J. D. Powell. An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. Computer Journal, Vol. 7, No. 4, 1964, pp. 303-307.

1639

GROUND RULES FOR MODIFYING PADS

In this section of the program description, some suggested procedures for modifying the applied load models, control governing equations and boundary conditions in PADS are presented. Referencing of the formulation document Volume I, is used throughout the discussion and both FORTRAN and engineering notations are used depending on which better describes the situation. Implications for input are discussed in the last sub-section.

Since the trajectory module of PADS contains both the steepest descent and quasi-linearization modules, there will be some special differences in programming changes. For the most part, those differences are related to the following:

a. Q-L Optimal Control calculation ALGCØN - AL1 subroutines

b. Common block communication

c. Arc initialization

d. Need for first and second partial derivatives in QL module and only first partials in SD module.

1. Applied load model change.

   The addition or modification of an applied load model requires that the formulation and coding of the accleration vector, a, and its partials with respect to state, y and the in-plane control vector, u, be redone. The coding for this type of change is outlined in chart 1.

2. Control governing equations.

   A simpler modification to the program is the addition or modification of a governing equation as given for example in Section 9 of Volume I.

1646

Chart 1

Applied Load Model Modification

Description	Applicable Subroutines: Steepest Descent	Applicable Subroutines: QL	Comments
Acceleration vector and its partial derivatives (1st, 2nd and mixed) with respect to state y and in-plane control u.	ACCEL	APPLY	
State only dependent terms in acceleration vector (e.g. aero-coefficients that are dependent on MACH number) and their partial derivatives.	EQUA3	STATEF	
Control vector dependent terms in the acceleration vector and their partial derivatives.	VTNOP VT BEROCØ MAMECØ IMPULS	UTNOP UT AEROCØ MØMECØ MØMECØ	
ARC initialization logic	PRØPIN PRØPB FNTG	ARCIN	Burn time initiation etc. ARCDAT common block "BOOT" B.
Common blocks that may need changes	GENF AECØ3 ARCDAT	DYNA ARCDAT	

If an applied load model is added or modified, it is often necessary to change one or more of the governing equation's subroutines.

The general form of the governing equation is

$$K_i = K_i\ (y,u) = 0$$

In the program the preferred form of K_i is

$$K_i = K_i\ [y,a\ (u,y)]$$

where a is the acceleration vector.

Each of the options for K_i is programmed in a separate subroutine along with its partial derivatives, first with respect to the control vector and then with respect to state. The reason for including the acceleration vector in the K_i dependency is that this vector and its partials are already computed at the time K_i and its partials are needed. This convenience permits a simpler representation of K_i and its partial derivatives through the use of chain rules. For example, the explicit partial of K_i with respect to α is

$$\frac{\partial K_i}{\partial \alpha} = \frac{\partial K_i}{\partial a}^{T} \frac{\partial a}{\partial \alpha}$$

3. Boundary Conditions

The function or non-linear target equations and their explicit partials with respect to state are all contained within subroutine PDBC in the steepest descent module and its parallel counterpart, PDBCQL, in the QL module.

In order to add or modify a boundary condition, it is only necessary to supply the equation for the function in terms of common supplied state and state functions and derive and code the partial derivatives of the function with respect to the state vector.

4. Input Implications

The impact of program modifications on the data input interface can be very significant.

4-a. Applied load model input implications.

For applied load model changes, the following must be considered. Arc dependent input data is stored as an array on random file. This array contains 51 words including such things as aerodynamic reference area, curve numbers, rated vacuum thrust etc. (See ARCDAT common.) The representation of this data in the input routine, INEDIT, (see this volume), is a set of 51 subscripted arrays; each array having 20 words to correspond with the maximum number of arcs in the trajectory. If it necessary to expand the number of words in this array, first the address equivalencing in the /AA/ common block in subroutine INEDIT must be shifted. Then the new variable names added to common /AA/ and to the NAMELIST/XX/ statement set. Common block /ARCDAT/ should be expanded to include the new variable names and arc initialization coding (see chart 1) should reflect the larger size of COMMON/ARCDAT/. If fixed point flags are added and print out of the new data is desired, subroutine FXDAT should be amended. Also the construction of the random file 9 should be changed to reflect the larger size of COMMON/ARCDAT/.

If desired, the meaning of words in /ARCDAT/ can be shared between different models in order to save the trouble of expanding the size of the common block. This could be handled by simple equivalencing in the subroutines that are affected. These routines should include INEDIT, where equivalencing

and dimensioning new variables in COMMON /AA/ could be accompanied by the addition of the new variable names into the NAMELIST/XX/ set.

B. Governing equation input implications.

Governing equation options are regulated by choice of option flags contained usually in COMMON/ARCDAT/. Therefore, the interfacing coding changes will only appear in the arc initialization and control choice logic. The arc initialization is described in Chart 1. The control choice logic is found in the steepest-descent module in subroutines MODELA, and BLGCON and in NPLANE and ALGCON in the QL module.

C. Boundary condition input implication.

The addition of a new boundary condition has important significance to the boundary condition checking and set up in Program (CDC) or Subroutine (UNIVAC) SDINP and its subsidiary routines. The logic for printout of the name of the target should be added to SDINP. Functions LOMG, MOMG, MPSI, TOL, SOMG, SPSI, TOLPSI, need modifications to reflect the addition of a target with a variable number code greater than 36, (see Table 2.6-1 in Volume III), requiring a preset tolerance value, and possible having different input units than internal units.